THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Theodore C. Bentley

TITLE OF THESIS: A Method for Priority Switch on Ethernet

DEGREE FOR WHICH THIS THESIS WAS PRESENTED: Master of Science

YEAR THIS DEGREE GRANTED: 1984

The University of Alberta

A METHOD FOR PRIORITY SWITCH ON ETHERNET

by

Theodore C. Bentley

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Spring, 1984

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **A Method for Priority Switch on Ethernet** submitted by **Theodore C. Bentley** in partial fulfillment of the requirements for the degree of **Master of Science.**

ABSTRACT

This paper outlines a method for implementing a priority feature as a variation to the Ethernet CSMA/CD network system. This priority feature forces the network to switch to another protocol. A system using this method suffers a decrease in throughput in the general case but gains a guaranteed maximum time for delivery of a packet and in some specific applications can increase throughput. A series of experiments run on a simulation of this system is discussed and the resulting performance analyzed.

Ethernet has gained wide acceptance as a method of connecting computing resources in a distributed environment. Ethernet has many strengths. It provides complete connection for the nodes. The maximum throughput as a percent of capacity is high. Each node in the short run has a equal chance to transmit and the control of the network is distributed. However there are problems. Ethernet suffers from a lack of a guaranteed delivery time. The specification even states that a delivery may not occur. It lacks an ability to assign any type of priority to a node. Further the protocol tends to give first in last out service when there is heavy system demand.

Can these deficiencies be overcome? Analog measurement is now used to detect collisions in Ethernet. There is unused analog information available that can be used by all nodes to note all collisions and to time the system behavior after a collision. With this information it is possible to more exactly control the nodes. The new protocol can offer a guaranteed delivery time and can assign priority to particular nodes. The new protocol can be followed for a short or long period of time and then the network can revert to Ethernet.

Acknowledgements

I would like to thank my supervisor, John Tartar, for his guidance and support throughout this research. I would like to thank Louise Bentley, my wife, who read rough drafts of the thesis and contributed help in a great many ways. Donna Fremont read the thesis and contributed helpful suggestions. My parents at all points in this endeavor were encouraging and supportive.

The late Dr. Demetrius Zissos was especially encouraging. Dr. Zissos took time to examine, discuss, and explain ideas in a way that gave useful new insights. He will be remembered for his energy, enthusiasm and vitality.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. The Problem

A commercially successful method of constructing a local area network is Ethernet [3,10,27]. This is a low cost network that provides complete connection between all nodes. All nodes are treated equally and the network has distributed control. This network has very good characteristics under many load conditions as well as some undersirable characteristics.

Ethernet does not offer a guaranteed delivery of a message, only a best effort. The inability to assure delivery has far reaching effects. Because of the possibility of nondelivery, a node can not successfully signal other nodes to demand use of the coaxial cable. Under no circumstance can a node be given higher priority over other nodes because it can not signal other nodes when it needs priority. Equality of nodes is usually quite beneficial, but if the equality can never be altered, then the transmission of any type of priority message is not possible. The present signaling system of Ethernet can not indicate when priority is desired by a node. This problem arises in Ethernet due to the unpredictability of the protocol at certain points.

Priorities and signals have proven very useful in other branches of computing yet have not been implemented on Ethernet. Priorities for nodes or messages would be a useful characteristic on such a network since all nodes and all messages are not of equal importance to the network or the processes running on the network. There are local area network methods that provide priorities, yet these are unsuccessful commercially. Therefore, it is attractive to have priorities on Ethernet or some other broadcast baseband network that is similarly cost effective. This thesis addresses the problem of the provision of a priority system for a broadcast network that gives good performance to a network, has a high data transmission rate, has a flexible topology and would be inexpensive to implement.

1.2. Proposed Methodology

This thesis proposes a protocol enhancement for Ethernet that allows a signalling facility and the implementation of a priority system while not deteriorating the performance of the present protocol. The new protocol allows all current Ethernet software to run, yet sets a maximum delivery time for a packet sent with priority status. By picking up more information from the transmission medium than traditional broadcast networks, the new protocol offers better performance for delivery of priority items. The new protocol uses the added information to better schedule the transmission of priority packets. This protocol greatly decreases delay for priority packets, and with some workloads would increase throughput of the overall system. The proposed protocol is compared to the current Ethernet. Since the goal is a measured comparison of two systems, appropriate measures and measurement techniques are identified and carried out. In addition, a number of protocols put forward in the past that exceed the performance offered by Ethernet are discussed with regard to their ability to provide a priority service to a local area network [6,12,19,22].

Chapter Two explains the history, protocols, and performance of local area networks. The chapter considers the computing environment where networks are used to explain the desirable characteristics of a local area network. A model for discussing network structure is explained and then used as a means to explore how and why a network can be altered. Toward the end of Chapter Two, an explanation is given on how broadcast network protocols, Ethernet being one, have developed as a sequence from simpler to more complex protocols.

Chapter Three outlines the new protocol, explaining its interfaces and layers. Reasons for the decisions on components of the design are given. A short analysis of the expected behavior of the protocol is performed. An explicit list of alterations to the present Ethernet protocol is set out.

Chapter Four gives a synopsis of the available methods for evaluating networks. Simulation is used for the testing of the new protocol. A specific simulation to assess the performance of the new protocol is presented, detailing what is and is not included. Tobagi, Tannenbaum and Klein-

rock have all discussed measures to use on network performance [14,22,24]. Items selected from these sources, used to assess the performance of the simulated network running the new protocol, are specified. The last part of the chapter outlines a series of experiments to collect data on the behavior of the protocol.

Chapter Five presents data from the experiments and discusses four topics. First, the validity of the simulation is demonstrated. Next, the simulation data is used to compare the behavior of priority packets from both the new protocol and regular Ethernet. Priority and non-priority packet behaviors in the new protocol are then detailed. Finally, experiment results to show the difference between the priority net and an Ethernet as a whole is presented.

Chapter Six draws conclusions from the experiment results. A discussion of possible further work follows the conclusions.

Chapter 2

Review of Broadcast Networks

This thesis proposes a modification to Ethernet to provide some of its nodes with a priority service ability. To set a context for the proposal, Chapter Two reviews the past development of *local area networks, LANs*. First, the environment in which networks have developed is discussed to show network applications and the demands placed on them. Situations in which priorities would be an advantage in LANs will be outlined. Next, a conceptual model of networks is described and related to LANs and priority transmissions. This model is a tool for network consideration and is used in much of the rest of the thesis. The alternate types of networks that can be used in small geographic areas are reviewed to understand why most local area networks are broadcast networks. The final section shows the development and current state of broadcast networks with particular emphasis on Ethernet. Different broadcast networks' performances are evaluated. Each network's ability to provide priority service is considered.

2.1. The Need for Networks

A *computer network* is an aggregate of computers linked by means of telecommunication lines. Networks provide the solution to many needs in a modern computing environment. It is common for a single organization to have many computers and related devices. This situation necessitates the ability to send information from one device to another quickly and easily. A network allows the exchange of information between machines.

For many computing applications, bringing the computers to the users rather than the reverse makes sense. The way that this is effected is to put small computers in many locations thus giving people ready access to the machines. Although a small processor or work-station can handle many jobs, results are generated or data is needed that is stored elsewhere and thus to complete a process, communication with a different location is needed. Networks give the ability to deal with

information in spatially separated points. Networks can allow sharing of costly resources while maintaining privacy and allowing incremental growth of computing facilities [22].

Although the computers may be used on large problems, not everyone can have a large machine. Subdividing the problem and distributing the parts to a series of machines is a way to allow small machines to be used on problems for which a big machine is normally required. This however assumes the existence of a method of communication between the machines. Thus, in the search for a way to do distributed processing, one of the necessary components is a communication mechanism that will allow processors to exchange information.

These trends have been accelerated by the advent of cheaper processors and the proliferation of computers. Each computer or computer peripheral device attached to a network is a *node*. A node can transmit or receive information from other nodes. The interconnection of nodes is effected by a *telecommunication subnet*. The information transmitted may concern the network or may concern some application. When a node on a network receives information via its telecommunication receiver it processes the information and may either use it or ignore it. Occasionally, networks use the messages at two levels of information, one for a process running on the device at the node, the other for the network. A node connects to the network by an *interface message processor, IMP* [22]. The IMPs along with a telecommunication facility actually transmit bits from one node to another. The telecommunication facility may be telephone lines, radio transmitters, coaxial cable or a variety of other means.

Networks can have various *topologies*, patterns of connection. Originally, the pattern had a hierarchical structure with a master central node , the other nodes were subservient [23]. Such a topology is a star network. As time went on, each pair of nodes that needed to communicate had a telecommunication line attached between them. The pattern of full connection is very expensive in its use of telecommunication facilities. As a result, nets were developed that could pass messages through a chain of nodes to the intended destination node. *Store and forward* networks as these are now called are used frequently in long haul transmission.

The development of new topologies has been spurred by a need to connect the machines in a more equal and direct way, and a need to make the connections more fail safe, cost efficient and numerous. A tremendous number of logical connections are needed and the number of desired connections is increasing as the population of processors grows due to decreased processor cost.

The density of nodes in a small area may be quite great. A network with many nodes all in a small area is called a local area network. The geographical restriction placed on a local area network is that all the nodes be within at most a few kilometers of each other. A small space such as an office building would be the usual domain of a LAN [11,22].

A popular class of LANs are the *broadcast networks* of which Ethernet is one. The resources needed to operate a network are processor time to address, package and receive information; telecommunications time to transmit data; and memory space to hold data that is waiting for processing or transmission. A broadcast network node sends messages via some channel or medium to several other nodes. Broadcast channels eliminate the need for intermediate nodes to use any resources to store or forward messages. As a result broadcast network nodes do not need to use processing time or memory space to store and forward material, reducing cost. Ethernet is a broadcast network that uses a coaxial cable as its medium. The messages carried on the network may be of quite varied importance to the network or nodes. Setting a priority sequence for transmission on a broadcast network is quite difficult.

2.2. The Need for Priorities

Equal treatment of nodes or packets is in direct conflict with priority delivery of messages on a network. The idea of priorities has proven very useful in other areas of computing but it is not available on the most popular forms of LANs. Research on networks has been directed toward producing the greatest overall throughput with the shortest average delay for message delivery.

Network performance is often measured in the following way. *Capacity* is total theoretical ability to transmit data. *Capacity is measured in bits/s or bytes/s*. Throughput is how much data is actually sent via a network. This may be much lower than the capacity due to poor scheduling of

the network's resources. A desirable objective is to get the throughput as close to the capacity as possible. The capacity is the theoretical limit of throughput and is seldom reached. To make throughput as high as possible the net must be used evenly and highly at all times. Network demand is often described as *bursty* because it is not equally distributed over time. The need for transmission tends to come in concentrated batches then die off completely for periods of time. If the demand on a network were at a constant level from all of the nodes all of the time then network design would be greatly simplified. *Delay* is the time elapsed between the completion of the preparation of a message for transmission and the time of its receipt by the destination node. Delay on some packets has to be high if maximum throughput is to be achieved. Some packets may be delayed quite a while until there is a chance for transmission. Less delay is desirable. Maximum throughput is antagonistic to low delay. Protocols generally try to get a maximum of useful transmission time out of this communal transmission medium while at the same time trying to allow all nodes an equitable use of the network.

The use of a priority or signalling system can help to achieve a network design with a more valuable throughput than a net with no priority. On a network there are several situations where priority packets could arise or be useful. Priority service would cost the system in over all throughput or delay or both, but it may justify itself in other ways. Up to this time most network research has focused on providing overall lower delays and higher throughput. Priority situations can often be predicted and may deserve special treatment. Here are some examples.

1) Priorities could be useful on a net in control of a real time situation if the network is used for activities that are not real time as well.

2) A CPU transmitting to a terminal population could use a priority system. The terminals logically rate a lower importance. The CPU should be given a means of gaining a greater share of the networks telecommunication time than any one terminal to be able to reply to all terminals.

3) A node which if given high priority would enhance system performance such as an interactive graphics device that always sends several packets.

4) A priority system could help for speeding the communication of important messages generated by a part of a distributed application system running over a network.

In designing a network, a decision has to be reached about whether to offer priority service at all. If it is absolutely vital and cost is no object, then a point to point link for all possible combinations of communicating nodes is the best, fastest and most reliable solution. This is a massively expensive method. At the other end of the implementation spectrum is integrating the priority traffic with regular network traffic. Integration is the approach that will be explored in this study. It offers a solution that is much lower in cost and would be more readily implemented if a network was already in place than providing fuller physical connection by dedicated lines. It also leads to the development of a more general network behavior.

When the decision to allow priority nodes to exist is reached, there are two types of priority service that can be given for a node. Either all the packets from a node need priority service or only some packets need priority service. This discussion will deal with the case where only some packets need priority. This generalizes to include the case where all of a node's packets need priority since the "all" case corresponds to a node that sometimes needs priority tagging a long sequence of packets as priority deserving. Either a prearranged sequence of transmission must always be followed or a node must be able to signal other nodes of its intent to transmit a priority message.

2.3. The ISO Model

The behavioral rules by which a network runs are called a *protocol*. The protocol running at each node determines the behavior for each node and as a result the behavior for the overall network. A group of nodes connected by telecommunication lines with no protocol is as useful to a user as a computer with no software. Protocols deal with several different interrelated aspects of the network behavior. Different protocols running on different hardware address common problems. An ordered hierarchy can be used to describe a network's protocols.

The International Standards Organization, *ISO*, has defined a seven layer model to structure the description of networks. This model is called the Open System Interconnection Reference Model, *OSI [21]*. Its intent is to make networks easier to comprehend and to focus attention on particular sets of problems that are common to all networks. The ISO model is a general description of the software and hardware needs of a network. The needs are grouped into layers that represent levels of abstraction including the raw data stream exchanged by nodes. The model is also intended to structure networks that need only a few layers altered if part of a network was revised. The model supports modularity in the design and planning of networks.

The seven layers of the ISO model and their functions are shown in Table 2.1. Sources that discuss the model indicate that it is not followed by all network designs [11,22]. Frequently, layers one to three are provided by the telecommunication subnet and layers four to seven are provided by the host computer or device. Discussion of LANs usually focuses on the telecommunication subnet. The higher layers are assumed to come from another source and can be interfaced to the telecommunications subnet.

A number of functions can be handled at many levels. Error checking is one of these functions. It can be done anywhere from the second layer up and can also be done several times by subsequent layers. Breaking up or reassembling long messages can also be done at several levels to suit the needs or abilities of a network's parts. Addressing is often done by each layer as if it was talking to another layer at the same level. Allowing the possibility of several layers doing the same function repeatedly means that network protocols tend to be redundant. For problems such as error checking, this increases the reliability of the communication. This also increases the traffic on a network and thus decreases the maximum rate of signaling between the highest layers. The final effect of the redundancy is that modularity is preserved in the network so that very different physical layers can be used for the same transport layer. A balance needs to be struck between the benefits of the modularity and redundancy and the costs of extra information transmission.

ISO OSI Model Layers		
layer		example functions
7	application	user process visible to the end user
6	presentation	encryption, translation, text compression
5	session	establish/remove user connection
4	transport	host to host communication, establish/remove network connection, segmentation of long messages into packets, reassembly of messages from packets
3	network	control telecommunication subnet, routing of messages, billing of users highest level that a packet for retransmission needs to rise to
2	data link	node to node packet transmission, error checking to give theoretically correct transmission
1	physical	exchange of raw bit stream, medium of communication set, frequencies of signaling set setup and take down of transmission electrical and mechanical details of connection transmission speeds of signals

Table 2.1 Layers of the ISO Model

The most fundamental layer of the model is the *physical layer*. At this layer a network protocol fitting the ISO model specifies how nodes in a network physically communicate. For example, the voltages or wave lengths used to signal the exchange of bits are defined. The mechanisms that carry out the activities of the physical layer are hardware devices. There is very little software used within this hardware; therefore, this layer is often accepted as a given for a network. Network designers may not be able to alter parts of the physical layer if an outside organization such as the telephone company owns them.

Within a node the physical layer is interfaced to the next higher layer, the *data link layer*. If it follows the ISO model, the data link layer specifies how the two nodes exchange groups of bits and agree to exchange the bits. The layers above the data link layer effect the rest of the interface needed to make a network useful to a human or an application program. Each successive layer shields the lower layers from the upper ones so that conceptually one might regard the transport

layers (fourth layers) in two nodes, as dealing with each other rather than having to deal with six intervening but lower layers.

A group of bits exchanged between data link layers is called a *packet*. Packet size, packet format, and correctness of transmission are dealt with in the data link layer of protocols conforming to the ISO model. An Ethernet packet has the following parts: a header, a data section and a trailer. The header has the network addresses of the sender and intended receiver, and a type field. The trailer of a packet includes a *cyclic redundancy check, CRC*. A CRC is a bit field that is a mathematical function of the 1's and 0's of the rest of the packet. The receiver takes the CRC of the packet, calculates its own CRC for the rest of the packet and if it equals the received checksum, presumes the packet correct.

A network layer, layer three, is not usually needed in a broadcast LAN since there is no routing, or setting up of a virtual circuit via a series of intermediate nodes to be done. The broadcast reaches all nodes. Removal of the third layer reduces the processing power needed in a network. The other function of layer three, billing of users can also be ignored in most cases since a single organization owns the whole net and after installation the cost of broadcasting is negligible.

2.4. Priority in the ISO Model

The provision for priorities has to be done as an integrated function of all layers of the ISO model. If a level decides to tag a message as deserving high priority, all layers of the net below have to be capable of sending the message rapidly or the provision of priority is not possible. If a lower layer does not provide a priority mechanism or cannot be controlled to allow expediting a message, the layer may bottleneck a packet in a queue with other traffic at that point. The lowest two layers of a net can effectively do bottleneck a stream of packets. This thesis considers how to implement a priority scheme for the physical and data link layers. Most protocols are written so that the data link layer only treats the physical layer in one way. To implement rapid priority service it may be necessary for the data link layer to be more flexible in its approach to the physical layer in order to optimize different aspects of network performance with different loads and when

sending different types of packets.

2.5. Broadcast LANs

Many methods of networking can be used for LANs; however, Thornton, and Franta and Chlamtac all indicate that broadcast techniques are the most suitable for this application [11,23]. Franta and Chlamtac in their book give several reasons why broadcast networks dominate the scene for LANs [11]. According to them, the strongest points in favor of broadcasting are:

1) broadcasting gives total connection of all nodes,

2) the telecommunication techniques used with broadcasting give generally higher capacity than the alternatives, and

3) there is no overall effect on the network if a node fails and quits broadcasting.

Total connection means that any node can send a message to any other node directly. This removes the need for complex routing algorithms in the third level that are needed if a direct connection is not available. To establish direct connection with point to point transmission means that the number of links installed would approach the number of nodes squared. For even a moderate size network, this can mean hundreds of cables that are physically difficult to put into a building and costly. Point to point connection is thus eliminated from consideration for LANs.

Ring networks are useful for LANs but are not as immune to node failure as broadcast busses, at least in theory [11]. LANs can have large numbers of nodes. Reliability of nodes is a prime consideration. If one node in a ring fails the ring is stopped. For this reason from this point on we will consider broadcast LANs, rather than the full spectrum of LAN methods.

Broadcasting is usually done on coaxial cable. This medium is relatively immune to interference. Baseband and broadband broadcasting are both possible on coaxial cable. Baseband uses only one frequency and any sharing of the medium must be some form of time division multiplexing of the cable. Broadband broadcasting uses frequency division multiplexing as well as time division multiplexing. Again, according to Franta and Chlamtac, the broadband broadcasting method

is not as popular as baseband [11]. The reason is that a head node that repeats all transmissions is needed. Failure of the head node would shut down the whole system making the reliability of broad-band lower than base band systems. Transmission delay is also increased for broad-band nets since all messages must travel to the head node at the end of the cable before being reflected back to the receiver. Broadband IMPs also are more expensive.

Other less important reasons exist that make baseband broadcasting LANs worthy of selection. Extension of a broadcast network is easier than extension of other network types. Maintenance and diagnostics are also less elaborate. It is easy to install new taps on the cable and most buildings can most easily accommodate a bus topology. These advantages make baseband broadcast coaxial cable LANs the attractive. Ethernet and the other network protocols discussed from this point on are baseband broadcast networks.

2.5.1. Development of Broadcast Networks

In the literature, the origin of broadcast nets is inevitably traced back to Aloha, a network developed at University of Hawaii by Abramson [11,22,25]. Abramson states the problem he worked on in the following way:

> Given the availability of a fixed amount of communication capacity, how does one employ this capacity to provide effective communication from remote users to a central machine? [15]

Abramson emphasizes simplicity and elegantly designs it into his system.

Aloha is a broadcast radio network that works over distances of up to thirty kilometers. The broadcasts are done by FM radio. FM is not suitable for LANs because of the interference within buildings and because of security problems. Aloha can be compared to a telephone party line where every node is always listening (that is receiving), and often one or more nodes may be talking (transmitting). Potentially this would lead to chaos since if more than one node transmits, the resulting messages overlap and are therefore useless. When two messages are sent at once a *collision* is said to have occurred. Received packets on Aloha have their addresses and CRC's checked.

If both are appropriate to the receiver, the packet is accepted. Aloha and many of its successors rely on probability. The fundamental assumption is that the network will probably be used by only one node some of the time.

In pure Aloha, nodes transmit at will, even though there is a chance that some other station will also transmit at the same time. The result is that the transmissions are mixed and can not be separated. They are discarded by the receivers. The only communication between nodes on Aloha is the digital interpretation of the signal received and sent. The transmitter pays no attention to any possible information to be gained in any other way. Aloha works despite the blind transmitting of packets. It is interesting to note that later mathematical modeling and simulation attempts to mimic Aloha had problems [25].

Aloha is the bottom end of performance for broadcast methods. The maximum ratio of throughput to available capacity that can be obtained with Aloha is 0.18. This is increased by *slotting* the broadcast medium and having nodes only start to transmit at specified times. This has the effect of eliminating the chance of a packet colliding with both an earlier and a later transmission. Maximum performance rises to 0.38 throughput/capacity with slotting. Trying to load the net beyond this results in less throughput because more collisions occur. A net that has less throughput as demand rises is said to be *unstable*. The instability arises because the transmitting node does not adjust to overall system load.

Packet collision is still a feature of Ethernet but not of all broadcast networks. Such collisions waste the available capacity of the broadcast medium. Collision free protocols do exist. They are achieved by signaling between nodes or by preloading a set order of transmission into the network. If signaling is used to prevent collisions, the signaling costs the network capacity in some way.

After the work on Aloha, development followed on broadcast networks restricted in size to one to two kilometers, that is "LAN" size. The first development used to increase performance was to have the transmitting nodes be more aware of the network. *Carrier sense multiple access,*

CSMA, is the name of this network type. In a CSMA network, a node senses the broadcast medium before transmitting. This is *carrier sensing, CS.* Many nodes can broadcast on the same medium so *multiple access, MA,* is allowed.

If a node senses a transmission on the broadcast medium, the node waits for silence or *defers* before starting to send. The result is fewer collisions and better use of available capacity. This gain capacity use is made because the node had sensed something of network state. Its behavior is in part being controlled from outside, but not by a central authority.

Another idea incorporated into CSMA networks is random rescheduling of transmissions if a busy carrier is sensed. If two or more nodes want to transmit, sense the carrier as busy, defer until they sense silence and then transmit, they guarantee a collision and an unstable network. However, if upon sensing the carrier the node picks a retransmission time randomly from an appropriate range, the chance of a collision is greatly reduced. This works best if the chance that two nodes will pick the same restart time is low. Network performance is greatly improved by this technique. Less capacity is wasted on collisions, more is used for useful throughput and as a result delay for delivery of a packet is far less.

CSMA works best if the nodes are all close enough to each other so that a packet takes longer to transmit than the round trip travel time of a bit on the network. Unless packets are to be tremendously long this means the network length needs to be LAN size. Thus when broadcast networks are used in local area networks, the possible range of useful protocols increases because packet transmission time is long relative to the end to tend delay transmission time. Broadcasting over a distance that allows the sender to still be active while the earlier part of the message has reached the ends of the net and has even possibly had time to travel back, means that the transmitter can take corrective action if there is a problem during transmission. The sender can know that its packet did not get through and needs to be resent. The transmitter must be able to sense the transmission medium and differentiate its signal from others in order for this to be useful.

Tobagi extensively reviewed and analyzed the performance of CSMA networks [25] . They greatly exceed the performance of Aloha and in some cases have throughput performance as high as 0.98 of capacity. To achieve such high throughput the distribution of restart times has to be closely matched to the size of the population of nodes trying to broadcast and all packets have to be long. However to achieve such a high throughput, packets are likely to be delayed quite a bit because very wide spacing of random restart times will have to be used. A method of tailoring the distribution of restart times was the major contribution of Ethernet.

In 1976, R. Metcalf and D. Boggs described a network protocol called Ethernet which fitted the physical and data link layers of the ISO model [16]. Ethernet is a copyright LAN of Xerox corporation [28]. Metcalf and Boggs' proposal of Ethernet carried on the idea of broadcasting and made two significant improvements: a backoff algorithm and nodes that could detect overlapping signals on the broadcast medium. They also added one limitation to the network: the end-to-end transmission time on the network was to be, at most, half the time for a minimum length packet to be transmitted. This means that if two nodes transmit packets that collide, both nodes will transmit long enough to receive transmission from the other node.

In Ethernet, a node wishing to transmit senses whether the cable is silent. If the cable is silent the node starts to transmit its packet of data. While transmission is going on, the transmitting node checks to see that it receives what it sends. If it receives something different from what it sent, the node senses that its packet is in collision with some other node. This ability is the *collision detect, CD,* feature of CSMA/CD. CSMA/CD is the generic classification of Ethernet. Once the collision has been detected, the transmitter puts a short jam on the cable to insure that the packet is damaged. The node then stops transmission.

The transmitting nodes' ability to sense that their packets are colliding has many benefits. The node has learned something more of the network's state and also knows it must retransmit the packet. Further, it can count the number of collisions encountered while trying to send any one packet. It can terminate transmission and not waste transmission capacity.

CSMA/CD networks, including Ethernet, are really only a development in the enhancement of the abilities of the transmitter. In Ethernet's case, the transmitter is made so that it can monitor the transmission going on, not only before it starts, but also while it transmits. Enhanced capabilities in the transmitter make the idea of *distributed control* of a network viable. Distributed control means that there is no one device or node in the network that schedules the use of the network telecommunication facilities. Each node decides when to transmit. In Aloha, the nodes do this with no idea about the condition of the net. This is really no control at all. In Ethernet, the node is aware of the network medium before and during transmission, and it governs its behavior by this information. This is a very effective form of distributed control which only exists because of the sensing abilities and the close proximity of the nodes.

The counting of collisions encountered while repeatedly trying to send a packet is the other improvement that Ethernet made to CSMA. This counting allows an estimation of how many other nodes are likely trying to send at the same time. If the number of other contending nodes is known, the interval from which to select random retransmission times can be set accurately. The bounds for retransmission times are set wider as the number of collisions increases. Setting this interval correctly allows more optimal scheduling of retransmissions than can be achieved with a static setting thus cutting delay time and increasing throughput while still keeping collisions low.

Ethernet is a particular variation on CSMA/CD [22,28]. Metcalf and Boggs' proposed system has become one of the most commercially successful local area networks [2,8,10,18]. Ethernet's present behavior is specified in a document called the "Ethernet Protocol" published by Xerox, Intel, and Digital Equipment Corporation [28]. The following recapitulation of Ethernet's behavior summarizes what has been said about it and serves as an example of how physical and data link layers of a network are integrated with each other and higher protocol levels.

2.5.2. Ethernet Physical Layer

In Ethernet, nodes are connected by segments of coaxial cable from which all nodes can receive signals all the time and onto which all nodes can broadcast at 10 Mbits/s using Manchester

encoding. Manchester encoding is a self-clocking signal that uses voltage transitions as bit signals. Each bit may have a voltage transition at its leading edge. The bit value is signaled midway through the bit by a voltage change, usually from low to high for a zero and the opposite for a one.

In addition to bit transmission and reception, the physical layer supplies variables for collision detection and carrier sensing to the data link layer. The collision detection variable is turned on only if the node is transmitting at the time of collision detection. The physical layer senses all collisions on the net except collisions between two other nodes. The data link layer does not get informed of all sensed collisions. Carrier sensing goes on whenever there is transmission from any node detectable on the cable.

Figure 2.1 shows a diagram of the information available at different levels in a node while an assortment of transmission states exist on the cable at a transmitting node. The pattern of voltage in this diagram is used in several subsequent figures. The time units are arbitrary and each period lasts ten units. It shows in sequence a period of silence, a period of transmission by one node, a period of collision involving two nodes, a period of collision involving three nodes, a period of collision involving two nodes, and finally a period of transmission by only another node. The wave pattern for the periods of transmission by one node are the shape that a repeated pattern of 1's and 0's would have in Manchester encoding.

In Figure 2.1, CD Ethernet2 shows the value the collision detect variable would have in the data link (second) layer of an Ethernet node. CD Ethernet1 shows whether or not the physical layer can detect a collision. CS, carrier sense, has the same value in both the physical and data link layers.

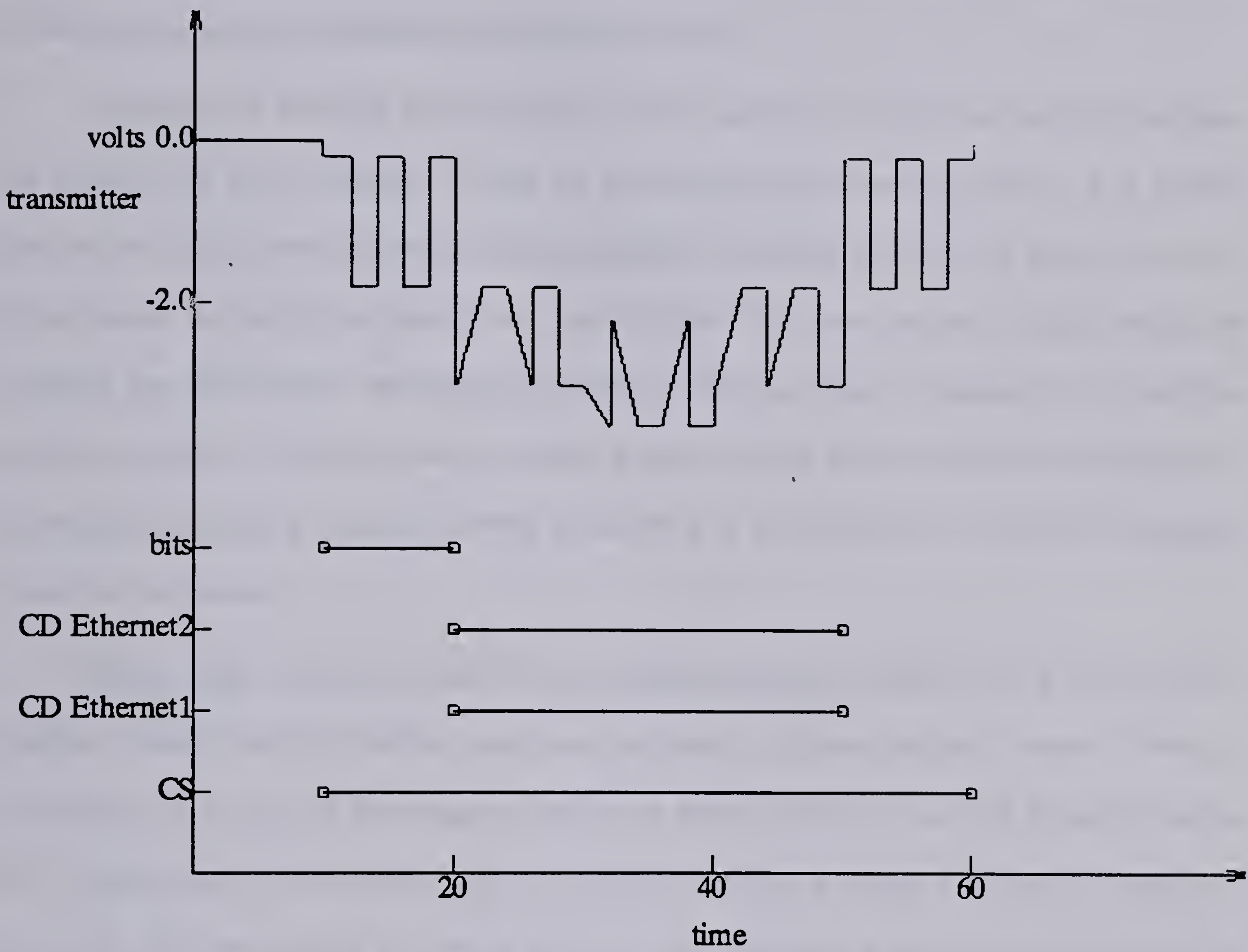


Figure 2.1 Information Available to Transmitter on Ethernet

2.5.3. Packet Transmission

Upon receipt of a packet from a higher layer, the data link layer checks if the CS variable is set. If not, the data link layer starts transmission of the packet by sending the first bit to the physical layer. The physical layer upon being given the first bit, sends a preamble to alert other nodes that a packet will follow and to synchronize the timing for reception of the subsequent bits. The bits of the packet are then transmitted in sequence as they come from the data link layer. Whenever it is transmitting, the physical layer senses for a possible collision.

If the signal received always matches the signal sent and the whole packet is sent, the data link layer informs the higher layer it has succeeded. However, if a collision is sensed, transmission ceases and the packet's transmission is rescheduled for later.

The sensing of collisions by a transmitter can be done in two ways. One way is to compare the bit sent with the bit received. If they are not equal there has been a collision. It is possible that two overlapping transmissions could have identical bit patterns and that they may exactly overlap so several bits may go by before two or more differ. The above process is digital sensing for collisions and would work well since the chance of two transmission matching for a long time period is very low. With this system, a receiving node can only detect a collision by noting that a packet failed to have a minimum number of bits in it or by detecting an error in the checksum which the jam causes.

Ethernet nodes could detect collisions and sense transmissions digitally, but in fact, their CD (collision detect) and CS (carrier sense) are set when a different effect is sensed. When a transmission is in progress the voltage on the line is always below zero and this is used to set the CS. When the node is transmitting and the voltage drops below a voltage threshold (1.7 volts) CD is set [9]. The lowering of the voltage occurs as each transmitting node adds current onto the coaxial cable.

The Ethernet specification states that the physical layer must sense all collisions between more than two nodes and between the node and any other node if it is transmitting. According to Crane, there is always a distinguishable voltage difference between one and more than one node's signal passing a point on the cable [9]. Voltage difference is used by Ethernet in the sensing of collisions by transmitting nodes. This is not stated in the specification.

Upon detection of a collision, the CD variable between the physical and the data link layers is set to true. The node takes the necessary action to reschedule a transmission in progress. Ethernet provides a simple, yet elegant method to determine when to transmit again. Because of the finite length of the network collisions will occur soon after transmission starts. This time period is

called the *channel acquisition time* and is 450 bit times. If a collision happens in this time, the transmitting nodes employ a *binary exponential backoff algorithm* to decide when to try again. The nodes whose packets are colliding keep track of how often their packets have collided. Then they pick a random number in the range from $1 \leq x \leq 2^{collisions}$ and try retransmitting that number of channel acquisition times later.

Figure 2.2 shows the information that is available at different levels in a node while receiving in the present Ethernet. It can be seen that the physical layer can detect collisions that the data link layer does not get information about. This is information about the state of the network and can be used in the scheduling algorithms for a protocol. If nodes that have not tried to transmit when a collision is sensed are programmed to back off and not try to introduce new packets into the contention, contention could be solved with fewer collisions and with a greater chance of the nodes getting first in, first out service. In Figure 2.2, CD Ethernet2, CD Ethernet1, and CS indicate the same things as in Figure 2.1.

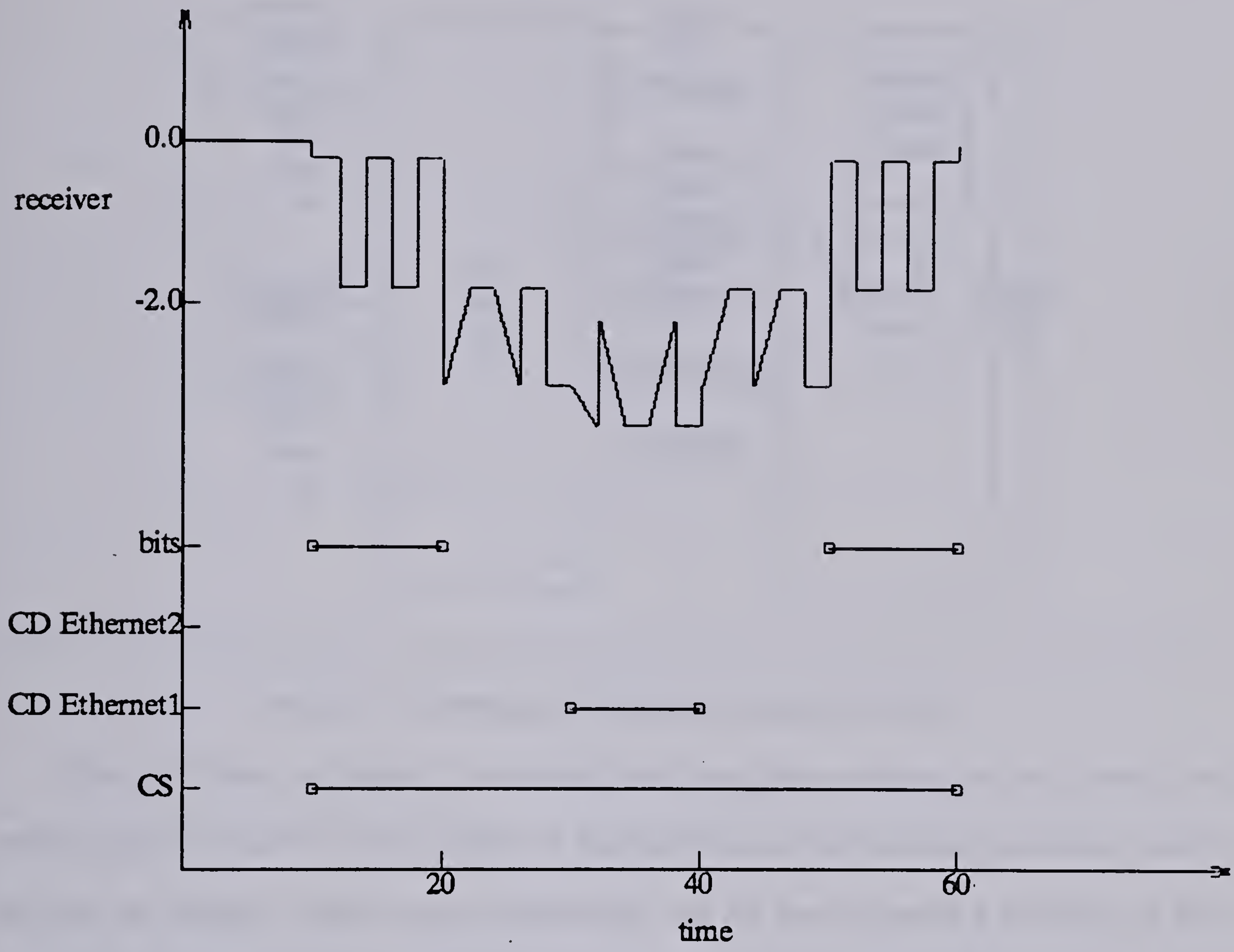


Figure 2.2 Information Available to Receiver on Ethernet

Figure 2.3 shows a block diagram detailing the variables at the interface of the layers of Ethernet. The neither block in no way represents the actual logical or electronic complexity of the structure. The coaxial cable would extend to other nodes. Note the "*and*" gate that ties the CD variable to the transmission variable. This is the actual arrangement of the wiring leading to the collision detect variable for the physical layer's interface. It explains why the data link layer does not learn of all the collisions that the physical layer does.

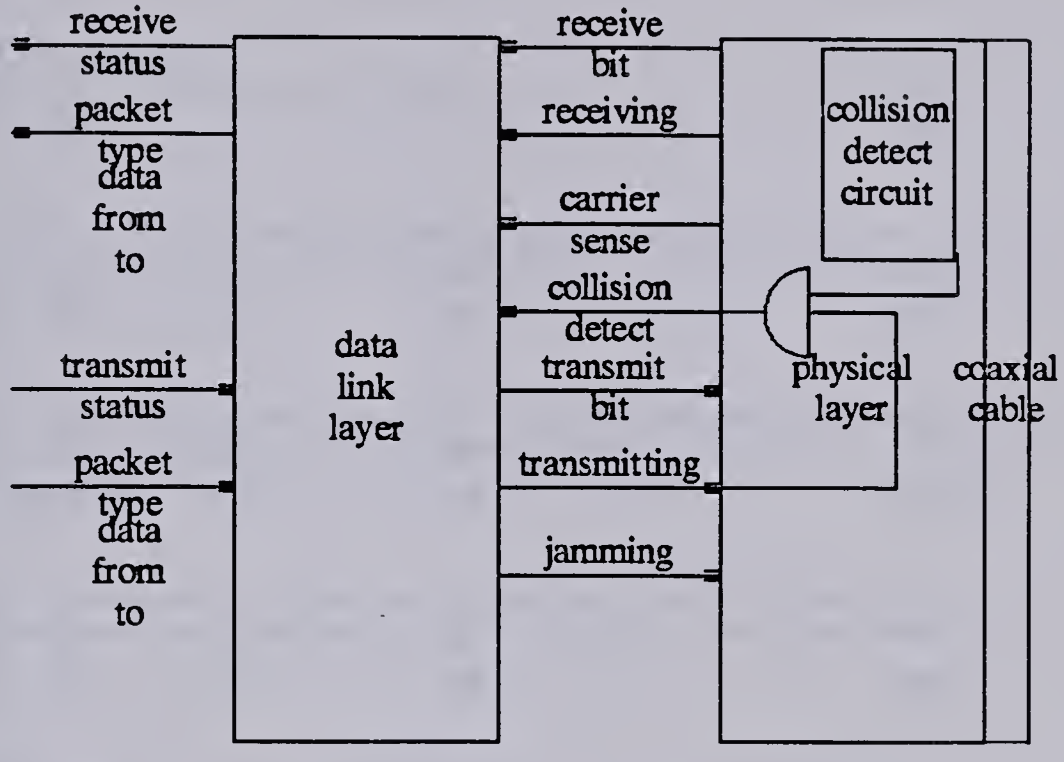


Figure 2.3 Interfaces and Layers of Current Ethernet

Figure 2.4 shows a diagram of a collision as it takes place over time for two nodes at the extreme ends of a network. The variables of the physical layer that indicate the network state to the node are indicated. The period of transmission from the two end nodes is a function of their separation on the cable. This diagram shows a worst case collision because it takes so long for the transmission of the nodes to reach each other.

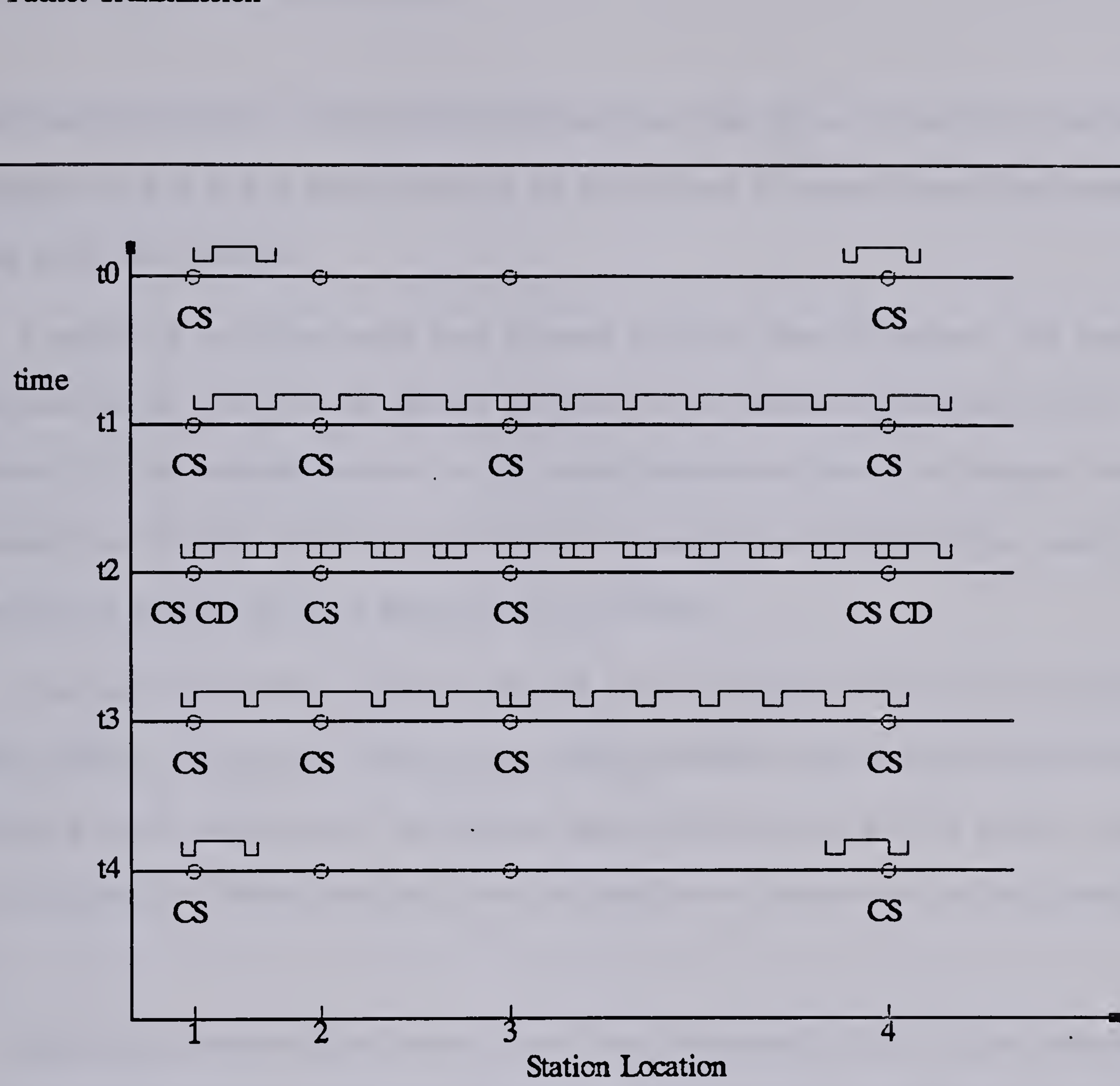


Figure 2.4 Network Response to Collision on Ethernet

2.5.4. How Well Does Ethernet Perform?

Many people have commented on the performance capability of Ethernet, in particular Blair, Shoch and Stuck [1,17,20]. The results can be interpreted to show it is either a good or poor system depending upon loading and configuration. Let us look at these studies in detail. Each used a different method of network measurement and each gives interesting insights about Ethernet.

Shoch's study was of a real Ethernet with 160 nodes at XEROX PARC. The nodes covered the range of possibilities from terminals to main frame computers. Franta and Chlamtac comment that this study is "a notable exception" to the lack of studies of performance of real systems [7]. Shoch reports on throughput, reliability, types of loading and frequency of collisions. The observed real system was never loaded above 0.40 of capacity. It was artificially loaded to 0.95 of

capacity and operated well. Unfortunately, Shoch gives no delay figures for the At low load, with throughput less than 10% of capacity, delay is the crucial factor to measure network performance. system at any load condition.

A number of interesting points were observed by Shoch about the system. The average packet size was 122 bytes. This was the only reference to a real average of packet size found in the literature. It is an interesting number to keep in mind when other systems are discussed from a theoretical point of view. If an average packet is 122 bytes, the usefulness of a great number of other pieces of research may be of decreased value due to the

A second point that Shoch observed was that almost all Ethernet packets were transmitted without collision. This indicates that the carrier sensing mechanism must be of great importance in maintaining system performance. The observed error rate was one in 2.0×10^6 packets. Shoch may be the source of Franta's belief that broadcast busses have a low error rate and high reliability [11].

Stuck did an analytic study of several types of local area networks [20]. No other analytic as opposed to statistical studies of Ethernet were found. The lack of other analytic studies appears to be because of the difficulties posed by the random restarting behavior of nodes dependent on the number of collisions. Stuck does not give any detail on his method of calculating the random part of a packets delay. The study does highlight a problem when comparing Ethernet to other systems. Ethernet has been specified to run over a long bus (2.5 km) and to interface easily with long haul networks. As a result, its packet format is longer than most other systems. It also does not assume infinitely small times for some aspects of transmission or reception. This high degree of specification causes problems when comparing it to other proposed methods of LAN because they do not have the practical considerations built into their specification. Stuck gets around this by adopting Ethernet's specifications for the other network methods as far as possible.

Stuck's study arrives at the conclusion that Ethernet is the best broadcast performer for loads that involve few transmitting nodes and a high data rate. He found that Ethernet is a relatively

poor performer for heavy loads coming equally from many users.

2.5.5. Ethernet and Priority

As was previously explained, priority treatment of some nodes would be useful on a LAN. Ethernet offers no possibility of priority service. Any node trying to transmit can collide with another and must back off. How often this would happen is not exactly predictable. Ethernet is specified so that if a packet suffers 16 collisions, the transmitter ceases trying to send the packet and reports to the higher layer that it cannot deliver. From Blair we also have the observation that first in first out service is not offered by Ethernet [1] . The service offered is a slight tendency to first in last out service. It is obvious that Ethernet, in its present form, is a long way from offering priority service. It does not even guarantee delivery.

2.5.6. Other Protocols for Broadcast LANs

In addition to Ethernet, a great many other protocols for local area networks have been suggested at least in theory. These have emphasized the reduction in the number of collisions and the increase in the possible throughput for systems heavily and equally loaded by all nodes. Research to optimize any other form of behavior has been limited. These studies give theoretical performance curves for the protocols. Comparisons are usually made to CSMA nets.

Time division multiplexing, TDMX, of a broadcast net is another method of scheduling. In it each node in turn is given a chance to send. At high loads if all nodes produced the same traffic volume, this system outperforms all others because all slots are used, no capacity is wasted on collisions or other signaling and throughput is at a maximum. Unfortunately this type of loading seldom occurs and TDMX is not adaptive to any other loading situation. The service it offers under other conditions is much worse than other methods. At low loads, usually a node will not send a packet and so its allocated time slot will be wasted. TDMX is not adaptive to system load. Consequently, TDMX is not used as a broadcast protocol.

A protocol that uses the idea of TDMX only for scheduling in order to remove the collisions of Ethernet is BRAM. BRAM is a token passing bus. This idea was suggested by two groups [6,13]. Each node in turn gets a small time slot in which to commence transmission. If the node does not start then, the next node gets a chance to start. Each node must have a clock to measure the time lag since the last transmission, and must keep track of which station last broadcast in order to know when it could start. The order of transmission chances is set when the network is set up and does not change. For multiusers causing heavy and equal loading, this system turns into TDMX since each node in turn would exercise its option to broadcast.

This system is known from analysis only, no examples were found in the literature of implementations. The system's performance is excellent under high load with many users. Although it avoids collisions, it does use capacity to schedule the net by needing time to transmit the tokens back and forth. When there are few users there needs to be one packet of scheduling time for each packet of broadcasting. The scheduling time can easily exceed the useful broadcast time. When one user is transmitting, no matter how high the load, one blank packet is sent for each useful packet. In an attempt to overcome the problems of the wasted capacity, there is a suggestion that groups of nodes be allocated to a single time slot. This would, however, lead back to the possibility of collisions and contention.

BRAM and its variants provide finite delay time for priority packets, an advantage over Ethernet. BRAM can not provide a way for a node to transmit ahead of lower priority nodes unless the priority node were given several broadcast slots in the sequence. This would deteriorate performance for all nodes all the time however. High speed priority service can not be achieved with BRAM while at the same time giving good regular performance.

Two protocols that are part way between Ethernet and collision free protocols are the Multiaccess Tree Protocol of Capetanakis and the Urn protocol of Kleinrock and Yemini [4,22]. These are limited contention protocols that seek to adapt between multiplexing the bus and allowing contention scheduling to go on as system load goes up and down. There is a window of nodes allowed

to broadcast. If more than one node tries to broadcast, the window size is cut to half and the first half of the nodes have an opportunity to broadcast. If a successful transmission occurs then the window moves to the second half of the addresses and they get a chance to broadcast. But if more collisions occur, the window continues to shrink into the first half of the addresses until only one node in the address window wishes to transmit, which it does. Then the window advances to the next group of nodes and a broadcast opportunity is offered. If no transmission goes on, the window grows and advances and a broadcast opportunity is offered .

This system is known from analytical study only. Its performance is very good if the adaptation of the size of the window can be made successfully at the correct times. The transmissions would also have to be evenly distributed throughout the windows. If the windows were of an optimum size and had nodes loading each window equally, the network performance could be optimum. Other loading characteristics would lower the network's performance. The protocol assumes that all nodes can hear all collisions and that they are able to keep track of the window size and location. The Tree Walk protocol has a worst delay equal to TDMX. It does not provide an easy way to provide priority. Each node must wait its turn and then broadcast, possibly after a number of collisions approaching the number of nodes.

One of the less fully discussed aspects of these protocols is how window adjustment goes on over a long time period. For example, if a set of nodes on a first round of broadcasts were within one window, do all nodes have to remember this for the next round and give the same window space again before changing window size? Can adjacent windows not on a power of two boundaries join? The use of memory and the use of processor time in this pattern of decision making could get quite extensive. The protocols also do not confront the practicalities of how all nodes are to sense all collisions. This would be needed in order to adopt the Urn or Tree protocol.

There has been an attempt by Spaniol to improve on Ethernet [19]. He describes an idea for priorities. While trying to keep fairly close to the present Ethernet, his changes call for a fixed packet size in order to synchronize some of the nets behavior. He also calls for some elaborate

timing activity to solve scheduling. The allowance for the new scheduling would cost several end to end delays between packets. Spaniol, is in reality, substituting a BRAM type of behavior that is only triggered after a collision. All this change is at the data link layer and the feasibility of implementation seems doubtful since all nodes cannot hear all collisions.

Gold and Franta have a method of broadcast networking that calls for more able receivers [12]. It calls for nodes capable of sensing all collisions. This method assigns each node a priority and by varying restart times implements priorities. This system would give low delay times for the highest priority nodes. Nodes could not choose between priorities for the packets they emit. There appears to be the possibility of one node of highest priority locking out all other nodes. All scheduling would result in the use of at least two end to end delay times. If the network was not ordered physically from end to end in a high to low priority fashion then the number of end to end delay times before the minor nodes were able to transmit would be long. The resulting loss of capacity would be severe. There does not seem to be a good balance between the use of priority and the offering of a useful general packet delivery system. A node that has a priority ranking must use it to send any packet even if it was able to judge that the packet did not deserve special treatment. This system has not overcome all of the problems of providing a mix of priority and nonpriority service satisfactorily.

No available method of broadcast networking has effectively dealt with all the problems to result in a system that provides fast priority service for nodes that need it and still give reasonable service to the rest of the node population. Ethernet is operational but has no priority and poor characteristics under a heavy load from many users. Other systems are not implemented and, or do not perform as well as Ethernet for single users. None of the systems really offer an attractive priority feature.

There is still room for a protocol that performs as well as Ethernet on low load, and high load with long packets or high load with few users, but adds on a priority or improves the behavior at high loads with multiusers and short packets. The focus of development of networks that can

put through the maximum number of bytes in a fair and equal way, assumes that all nodes are equally likely to be active and that each node will contribute equally to the network use. This is unlikely to be the real situation on a network. The nodes of networks are tending to get more diverse, not more equal. What would meet the need of these diverse populations of nodes is a network that can offer a more finely tuned treatment of different categories of nodes.

Chapter 3

Design of New Protocol

The present question in designing a broadcast network is: given the availability of a fixed amount of communication capacity, how does one employ this capacity to provide effective communication between members of a population of diverse nodes. This chapter will explain a protocol that retains the good aspects of Ethernet but has added to it an ability to give priority to some packets. The protocol explanation is done in several stages. Each successive stage gives greater detail on the parts of the protocol.

Ethernet was chosen as the base protocol because of its commercial acceptance, its generally good performance and the fact that it has been implemented and run. It has some disadvantages but is a major broadcast technique.

3.1. General Goals

The protocol presented here is an extension of the present Ethernet. This protocol retains Ethernet's very good performance for a high load caused by a single user or low system load, and adds a priority message feature with low delay at any load. The delay for a priority packet should be less than that for any other broadcast method. The other objectives are:

1) to maintain software compatibility with Ethernet;

2) to maintain the distributed control of the system;

3) to avoid having one node take all the transmission time due to excessive priority ranking;

4) to provide a design which is technologically possible now;

5) to provide as high a level of service as possible for the nonpriority packets;

6) to allow a node that can send priority packets and can discriminate between packets the option to send regular packets and avoid over use of the priority feature;

7) to keep the maximum delay of a priority packet in the same order of magnitude as the average priority packet; and

8) to assure that a high priority packet does not have any possibility of first in last out service or nondelivery as with Ethernet.

At the most abstract level, the protocol offers regular packets an Ethernet service, unless the coaxial cable is being used for priority transmission. High priority packets are offered a type of service similar to BRAM. It is presumed that nodes of differing capabilities and uses exist, and that the ones needing priority service are a small set of the total population.

When a packet is transmitted, the protocol initially tries sending all packets as Ethernet packets. However, after its first collision a priority packet gets different handling. The transmitting node, by means of the jam signal sent on the first collision, notifies all other nodes to switch to priority mode, and the packet is transmitted having suffered a maximum of one collision and having waited only for other priority nodes ranked before it which had packets to transmit. After a round of priority transmission, the network goes back to Ethernet.

The mechanism that allows implementation of the necessary signals in this system is a greater use of information from the coaxial cable allowing a node to indicate to all other nodes when to follow a priority protocol. The motivation for combining these two network protocols is to remove the random behavior of Ethernet. The priority protocol allows groups of equivalently important nodes to time division multiplex the bus. Nodes with a priority status can still send a regular Ethernet packet and therefore not abuse the priority rank.

3.2. Concepts of Changes

To add a priority feature to Ethernet, the major problem is making it reasonable to assume that all nodes can receive a message to defer when a priority packet needs to be sent. This must be done in an unmistakable fashion by an asynchronous signal that travels to all parts of the network in order that all nodes can sense it and alter their behavior accordingly; otherwise, distri-

buted control is lost. It can not possibly be done with digital signaling on the present Ethernet since no packet is guaranteed delivery. Not even if priority nodes were " *one persistent*" about transmitting would this work. *One persistent* means that a CSMA node always tries to transmit whenever it has a packet to send and senses no transmission on the cable. If two priority nodes which were one persistent had packets that collided, they would always collide. In short, for the same reasons that Ethernet can not guarantee message delivery, it can not send priority messages as digital signals.

The solution to this problem in signaling is to generate a nondigital signal. To do this, two things must be possible. First, all nodes must be able to detect all collisions. Second, all nodes must be able to measure time after the start of a collision. Given these two things, it is possible to signal the type of packet to all nodes if transmitting nodes give off short and long jams according to packet type. Collision sensing and jam timing by receivers make the protocol possible. Minor changes to the physical layer and some alteration of the data link layer will do this. The Ethernet's behavior can then be altered on signal so that a behavior that gives predesignated priority to nodes can be followed. A node wishing to push through a priority packet that suffers a collision needs to set up this new order in the network. The priority half of the protocol is one that does not use or allow random restart times to be scheduled. It removes the random behavior of Ethernet.

In the new protocol, the receiving nodes are more sophisticated in the sense that they are able to detect and time all collisions that occur on the network. Collisions, as mentioned in the reference to Crane in Chapter 2, are currently detected by sensing voltage levels on the coaxial cable. Thus, if a jamming node were not just to put a few random bits on the cable as a jam but were to do so at a higher than normal current, all receivers could sense all collisions because of the higher resulting voltages on the coaxial cable.. All receivers, in the new protocol, sense collisions and respond to them by starting a timer. The timer runs until the receiver can no longer sense voltage on the coaxial cable. The reading of the timer when silence returns is then evaluated to deter-

mine if the jam was to call for priority mode. When jamming, the transmitter needs to cause slightly enhanced voltages for jam bits and the receiver sensing this higher voltage needs to set a timer.

The reason for adding the timer to the receivers is that the ability for a receiver to sense all collisions would not, in itself, enable a priority system. A post collision timer allows the times for collisions to be divided into two groups on the basis of length. A node can thus distinguish a regular collision (which takes less that a round trip time) from long collisions (which would take more than a round-trip time). The longest possible time that carrier sense could be on after a collision with Ethernet is slightly less than 450 bit times. This would be caused by the round trip delay for a node at the extreme end of the net having its signal collide with the signal of a node at the other extreme end of the net. Collisions that last longer than 500 bit times are caused by a different phenomena. A collision of greater than 500 bit times is a call to use priority mode. Given differential collision times a node can decide if a call for priority is being received.

The transmitter is also slightly changed for this protocol. The jam put out upon collision is no longer just a few random bits to ensure a bad CRC. The jam is at a slightly enhanced voltage so all receivers will pick it up. This is needed since with regular voltage, collisions are possible that would result in nodes who were not trying to transmit not receiving the high voltage of a collision and regarding the transmission they do receive as just a packet fragment.

Since the address or addresses of the priority seeking nodes can not be discovered from the collision or its aftermath, the subsequent scheduling of the network must be set in advance. This schedule is prearranged for the data link layer of the protocol. When a call for priority is received, the nodes then change to a token passing bus. All nodes that have a priority status in turn get a chance to transmit packets. All other nodes remain silent until the net switches back to Ethernet behavior.

If a node with a priority status has no priority packet queued when its turn to transmit occurs then the node must signal that it will not send a packet so that the other nodes waiting will proceed

with their turns. The node with no packet puts a token transmission on the cable to signal that it is passing up its turn. A token bus effectively makes the nodes TDMX the bus, allowing short time segments for nodes with no packets, and time segments long enough to transmit a packet for nodes a high priority status.

3.3. Specific Changes

The following specific changes to the Ethernet Specification are needed [28]. The physical layer receiver needs to have the "*and*" gate that ties its *collision detect* variable to the *transmitting* variable removed. This allows the data link layer to be informed of all detected collisions. The physical layer transmitter needs to transmit a signal causing a higher voltage when *jamming*. The interface between the physical and data link layer needs a *jam* variable added to it since jamming is currently just an extension of transmission. The data link layer needs a timer to time jams and jam reception. This could be the current clock used for timing of interframe spacing and backing off. The data link layer needs the software for a token bus. The data link layer to higher layer interface could be unmodified, however in the protocol simulated the interface has a bit added. This bit informs the higher layer whether or not a packet it has presented to the data link layer is actually being transmitted. If it is not, the higher layer can substitute a priority packet for an already presented Ether packet. A transmitting data link layer knows when a packet is to be given priority status as the type field in the queued packets sets this.

When the network is brought up each node must know how many nodes may use priority. If the node being brought up is a priority node, it must know when its turn in the sequence of nodes with priority transmission opportunity is. The priority order in the network can be reconfigured while the network is running provided that it can be verified that all nodes receive the reconfiguration message. This can not be done at the time that some node wishes to used the priority system since massive problems with controlling the network would result.

The two different lengths of jam are controlled by having the node emitting a jam time the jam length. The length of a short jam must be less than the length of a long jam minus an end to

end transmission time. One workable combination of jam times is five bit times for a short jam and 500 bit times for a long jam.

The over all effects of these changes are illustrated by the following diagram for transmitters (Figure 3.1), and receivers (Figure 3.2). The transmitter's level of network state knowledge is essentially unaltered. Upon collision, the transmitting node's receiver is more active than for Ethernet so that a transmitting node gets a timing of how long carrier is on the cable before silence returns. For the receivers, the data link layer gets some new knowledge. The removal of the "*and*" gate in the physical layer enables information on all detected collisions to be passed up to the data link layer. The receiver of all nodes especially the quiescent nodes has suddenly gained a great deal more knowledge of the network at both the physical and data link layers.

In both diagrams, the lines for the regular Ethernet are included with the subscript of Ethernet. The lines for the proposed protocol have the subscript "new". The data going by is the same as that used in the examples from Chapter Two (Figure 2.1, Figure 2.2). The CS line stands for carrier sense at both the physical and data link layers for both the new and Ethernet protocols. The CD line stands for collision detect. The 1's and 2's indicate at what level the information is available.

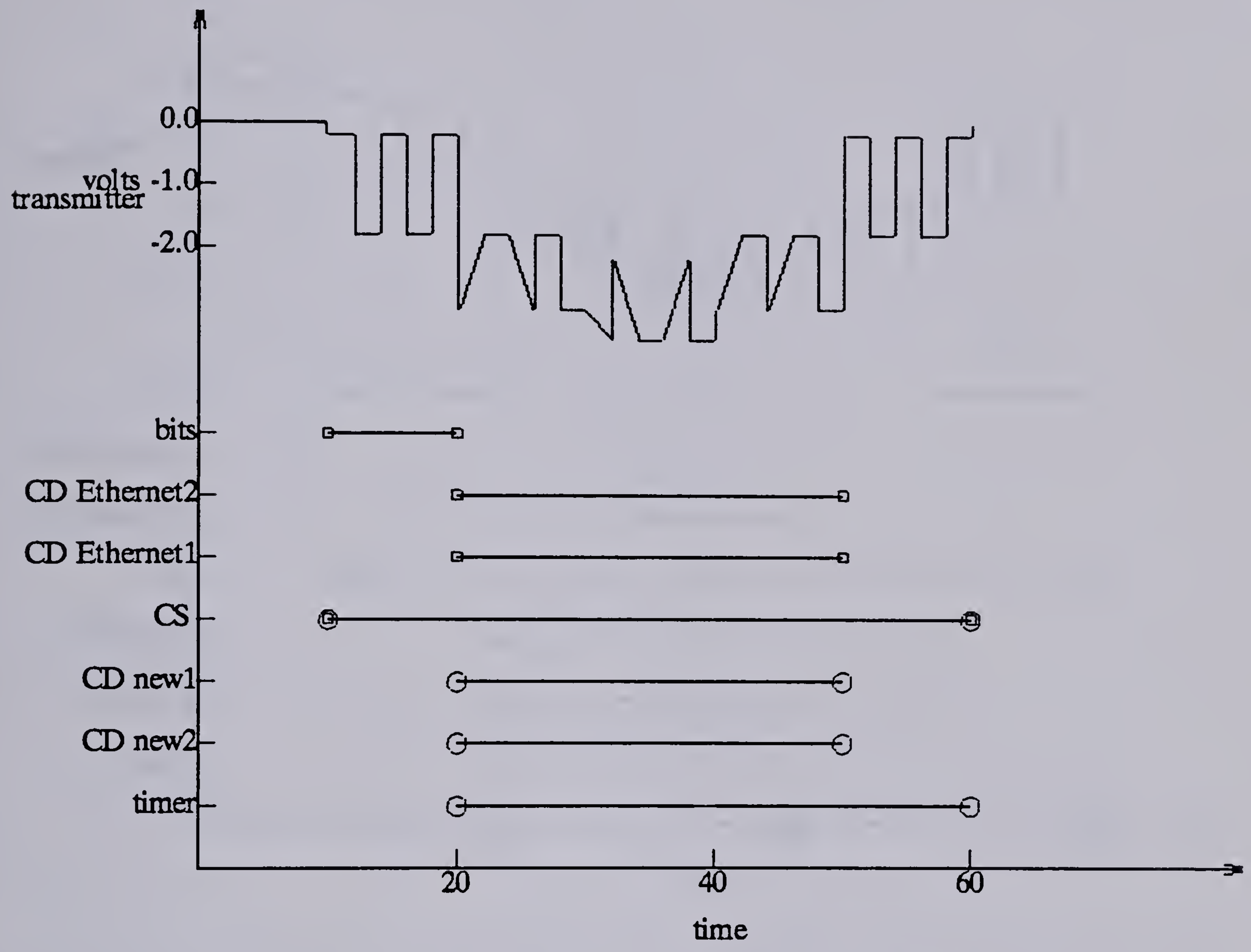


Figure 3.1 Information Available to the Transmitter New Protocol

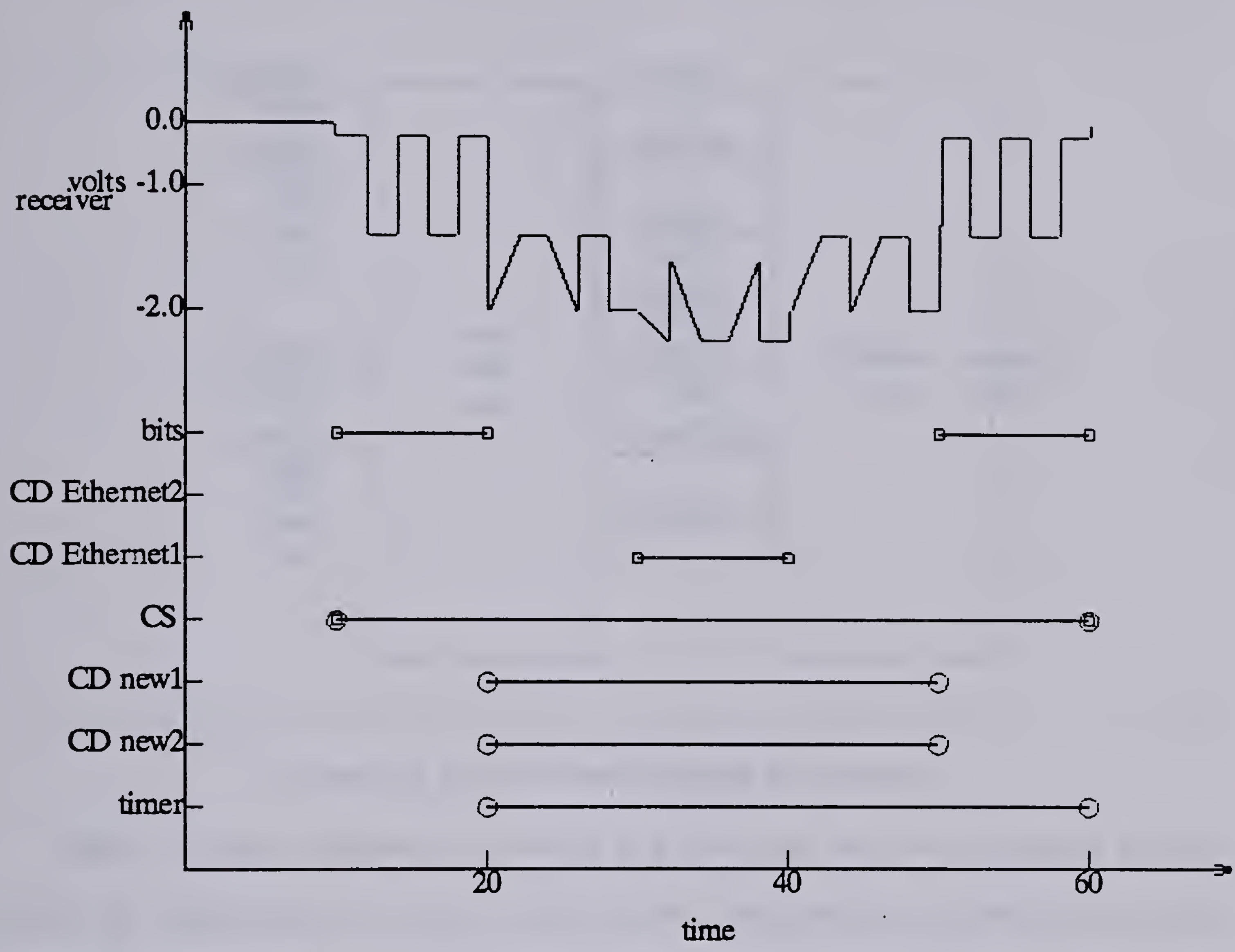


Figure 3.2 Information Available to the Receiver New Protocol

Figure 3.3 shows the interfaces and layers of the new protocol. A jam variable has been added to the data link to physical layer interface to allow the data link to control the voltage as well as the length of the jam. The "*and*" gate that ties the CD variable to the transmission variable has been taken out of the physical layer.

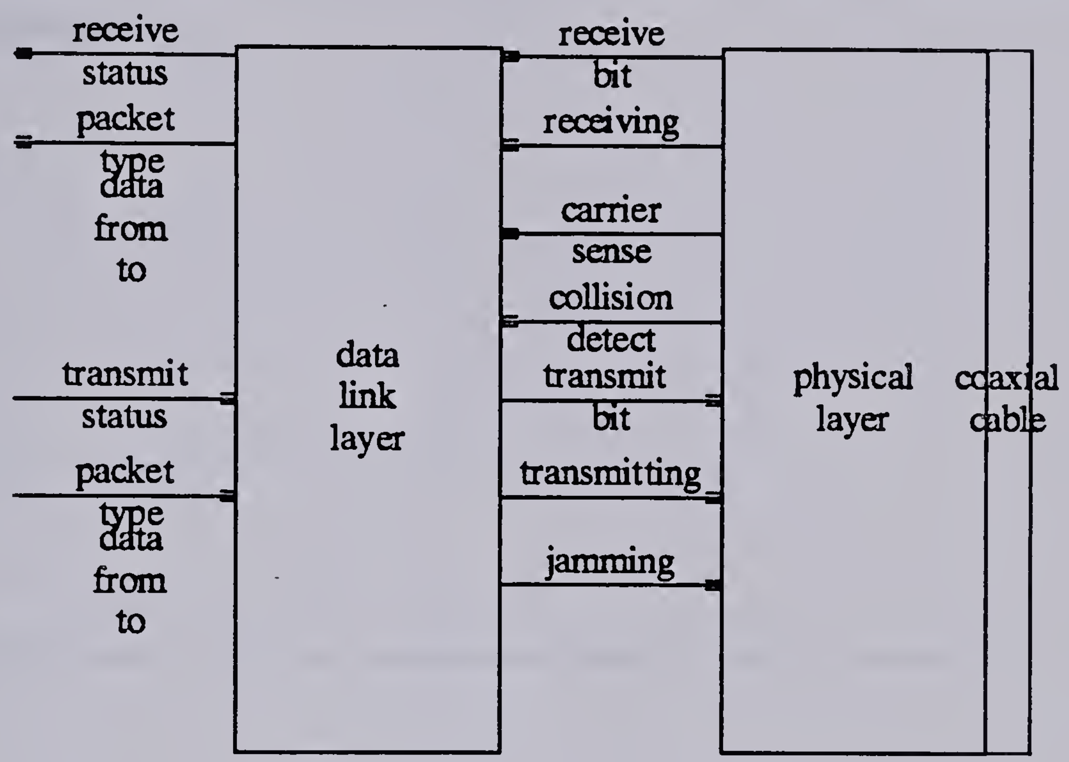


Figure 3.3 Interfaces and Layers of New Protocol

Figure 3.4 shows a diagram of a collision as it takes place over time for packets from two nodes at the extreme ends of a network. The variables of the physical layer that indicate the network state as sensed by the node are indicated. The intermediate nodes are more aware of the state of the rest of the nodes and the network in this protocol than they are in Ethernet (Figure 2.5).

If a physical layer such as the one just described is available, users can chose between several protocols. This physical layer can support Ethernet, BRAM, Urn protocol, and TDMX plus many combination of parts of these protocols.

The written descriptions of all the above mentioned protocols, except Ethernet, do not clearly specify a physical layer. The barrier to their commercial use may, in fact, be lack of technology. This physical layer would work for all the above protocols and in the simple form of a broadcast coaxial cable. The physical layer just described has an ability to sense all jams can give BRAM the physical layer it needs to work. The Urn protocol could be practically implemented since all nodes

would hear all collisions and could therefore set their windows appropriately. This physical layer would allow a system designer much greater latitude to design or select an appropriate protocol for a particular environment.

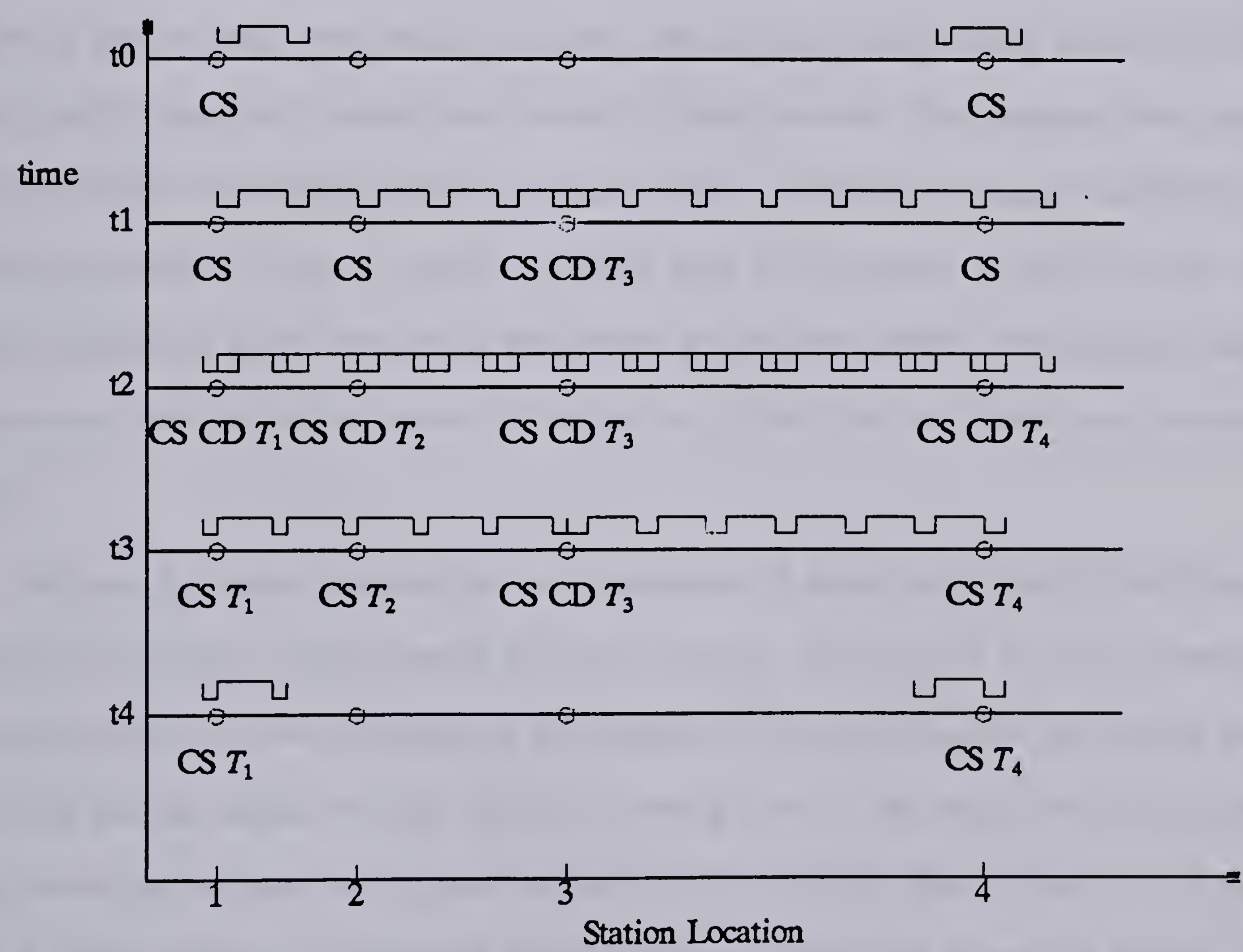


Figure 3.4 Network Response to Collision in New Protocol

3.4. Expected Performance of the New Protocol

The performance of the new protocol described in this chapter should give low delay for priority packets, slightly higher delay for regular packets and allow high throughput for the whole network. For priority packets, the protocol sketched removes the uncertainty about delivery that is inherent in Ethernet. Assuming that only a few nodes on the net were given priority status, the feature would give good response for priority packets. A priority packet would experience little delay since the population of nodes that can gain priority is limited. The provision of a short and predictable delay is essential for a priority message system.

The maximum delay of a priority packet once it has started transmission on the bus is easily calculated. It is the round trip delay time plus the number of priority nodes times the maximum packet transmission time. The time spent queued in the transmitting node varies depending on the number of packets being generated by the node. The maximum delay during transmission rises linearly with the number of tokens given out and is clearly bounded. The maximum delay is also easily controlled by limiting the number of priority nodes. When token passing is established, all nodes expect there will follow a number of packets equal to the number of priority nodes. If a priority node always needs to be able to send several packets for a priority transmission, it can be given several tokens when the network is brought up so that it can in sequence send a burst of packets.

The delay for regular packets is not easily calculated. It would be the delay for an Ethernet packet plus some added delay caused by the priority scheme. The delay for the whole system can be pushed either up or down depending on the proportion of the load presented that requires priority service and the number of nodes the load is coming from. If the whole load is equal from many priority nodes, delay will go down because TDMX will be in effect. However, it is more likely it will be pushed up somewhat by the extra deference required of nonpriority packets. The throughput of the whole system will be slightly less than for Ethernet since part of the capacity is needed to transmit the longer jams.

3.5. Comparisons and Contrasts to Other Systems

The priority scheme must also be compared to other forms of broadcast networks. Priority packets will have less delay than packets on BRAM because of the smaller population able to access the net. Priority packets will also have less delay than the tree walk protocol for the same reason. Franta's new protocol will, for some nodes, give better response and less delay [12]. However, the delay of the overall system can be much worse. Several round trip delays will occur before the low level nodes can transmit and their contribution to the overall system delay will be great.

To summarize, the new protocol will run exactly as an Ethernet until a long jam period follows a collision. Packet format, preamble generation and interfaces between layers are all preserved. If a long collision is sensed, the net will change to a token passing bus with the priority nodes each in turn broadcasting a packet or putting on a mini packet (token) if they have no packet to send. When all priority nodes have had a chance to broadcast, the network returns to Ethernet. Each node, in addition to being able to behave as an Ethernet node, can detect all collisions up to the data link layer, and measure the time from detection of a collision until silence returns to the cable. In addition, if a priority length collision occurs the node either waits its turn to transmit, or waits until all priority nodes have transmitted before it returns to Ethernet protocol.

Chapter 4

Simulation

Simulation is only one of three possible methods for studying networks. This chapter explains why simulation was chosen as a method of studying the proposed protocol and then outlines the method of simulation. Advantages and disadvantages of simulation in this situation are discussed. A description of a simulation to test the protocol is set out. The description includes a section discussing what network measurement capabilities are built in to the simulation and what aspects of the real system are not modeled. The last part of the chapter lists the experiments carried out using the simulation. These experiments show that the simulation is valid and show the behavior of the new protocol.

4.1. Methods of Network Study

There are three major methods of assessing networks, an explanation is in order to show why simulation is the choice in this case. The other choices would be construction of a prototype for the network and mathematical analysis of the proposed protocol. These three methods form a spectrum from the practicality of a concrete object to the theoretical abstraction of a model. Construction is at the concrete end, simulation is in the middle and analysis is at the abstract end.

Construction is the ultimate test of a network protocol. It is the reality that theory seeks to predict and inevitably provides the maximum resolution of detail for behavior. Construction does have drawbacks, especially for the research environment. Observation of the behavior may not be easy or in some ways even possible. For example, if precise time measures are sought for physically separate nodes, it is unlikely that precision can be achieved. There are problems of communicating between measurement points when measuring time precisely in a constructed network. Measurement of the time of an event's occurrence with an accuracy of microseconds at several spatially separated points is technically difficult. This type of measurement would be needed for some

observations of networks. Another problem with network prototype construction is economics. Purchase of electronic components is costly. Use of machine time to drive a network, if it is available, could approach machine time needed for study by other methods and this is costly. Software, because of detail, may also be more time consuming to construct than simulation or analysis and thus more costly. Construction requires technical expertise to plan and assemble the electronic hardware. In the end, the results generated for a researcher may be no more valuable than results generated by less arduous methods. Reports that systems have been constructed are not rare, but reports of well measured performance are. One report of real measurement of Ethernet was found [17].

Mathematical analysis of network performance is certainly the most frequently reported type of network assessment. Many studies performing analysis were found for LANs [12,13,19,20,25]. However, almost all refrain from analysis of Ethernet, even though it is more frequently mentioned and advertised as a product for sale, than the methods analyzed. The one analysis of Ethernet that is presented plots points only for the most extreme situations [20]. It does not explain in detail its method of calculation of delay for delivery when more than one node is trying to transmit. Analysis of a network running a protocol that asynchronously multiplexes a transmission medium and that has each node varying packet sizes and probability of transmission is a major undertaking and was not attempted.

In a choice between analysis and simulation the choice is between ease of mathematical representation and ease of behavioral representation as offered in a computer language. In this case, computer representation is easier and yet providing more detail than other available methods. Another factor is that what is being tested is largely software. The data link layer, although it has been done in hardware was originally software [10]. The physical layer can be represented in about 180 lines of commented C code so it is small. The two problems with computer simulation of the protocol are adequate representation of the coaxial cable and generation of a reasonable load of messages for the network to transmit.

4.2. Simulator Design

Zeigler classifies the simulation carried out as a discrete time discrete event simulation [29]. It is a discrete time simulation because the minimum time period modeled for the coaxial cable is 10^{-7}s. Subsequently this is set to 8.0×10^{-7}s or 1 byte time. It is a discrete event simulation since no effort is made to model the electrical behavior of the coaxial cable in any detail. The simulation replicated the behavior of the data link layer. The interfaces to the higher levels and to the physical layer are also exact.

The simulator is constructed as modules of software divided in three layers to correspond to the ISO model of networks. The physical and data link layers are simulations of the system. The third layer is a driver and measurement device. There are two modules in addition to these three layers. A start-up software module reads in a configuration for the system including the nodes' types, levels of activity and positions. When the simulated time period has run out the overall system statistics are then calculated by a data summary module. Figure 4.1 shows a diagram of the major blocks of the simulation.

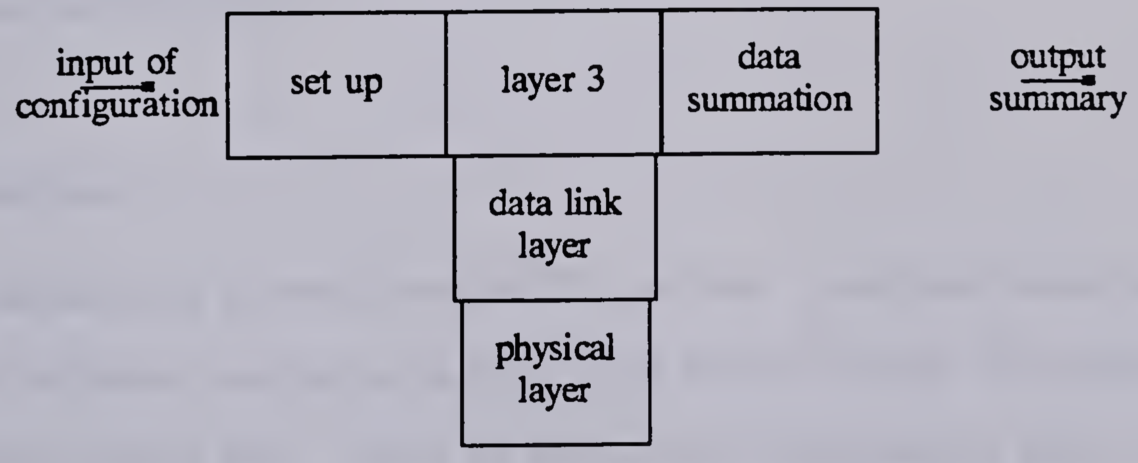


Figure 4.1 Major Components of the Simulator

4.2.1. Set-up Module

The simulator is designed to provide a flexible tool for measurements of two protocols: Ethernet and the proposed protocol. The simulator's behavior is controlled by the configuration which is input as the first step of a simulation run. The system variables that are set by this read-in process are:

1) location of nodes relative to their left and right neighbors, This is expressed as bits or bytes of separation. If the separation of two nodes is n-bits then they can transmit n-bits on the coaxial cable before the leading edge of the first bit passes the tap on the coaxial cable of the other node.

2) the node's addresses for Ethernet packets, and multicast packets as well as its token number for priority if any,

3) the number of nodes to be given priority status,

4) the message to send in the packets, this inevitably is Jabberwocky [5],

5) the mean arrival time for packets sent with priority or Ethernet status from a node, The arrival times are generated by drawing random values from an exponential distribution of inter-arrival times with a predetermined mean.

6) the size of packets to send from each node,

7) the duration of the experimental run in microseconds of simulated time and

8) total number of nodes.

4.2.2. Third Layer

The simulation has no layers above the ISO layer three. Layer three creates packets with an exponential distribution according to the mean arrival time for the node. The packets are sent to other randomly selected nodes. Part of the address field of each packet is used to transport the time of origin of the packet. This allows a receiver to calculate delay for a packet upon its receipt by subtracting the starting time from the receipt time. The delay total for all packets, number of

bytes of data sent and received , and the number of collisions as well as the maximum, average and variance for these measures are kept as running totals by each node's layer three. The measures taken are more fully discussed in section 4.2.6 (Measurement of the Simulation).

With this one system, both Ethernet and the new protocol can be simulated. If all nodes are loaded with zero percent priority packets the system simulates Ethernet. If it has some nodes giving out priority packets, it models the priority system.

4.2.3. Data Link Layer

The simulation of the data link layer includes the full detail of the level 2 protocol. It has the interfaces of an Ethernet data link layer and can be made to behave as an Ethernet data link layer. The only data link function not performed is the calculation of a CRC. Coupled with this, there is no simulated error rate as it is noted in Chapter 2 by Shoch that the error rate is low. Other items such as the reception and transmission of bits, packet reception, packet length checking, and backing off are as set out in the Ethernet Specification [28].

The priority software is added into this code. The major function added to the existing software is a mechanism to count and keep track of tokens during token passing priority bus operation. This means a change in the deference procedure as well. The jam procedure is also changed to allow a node to jam for two distinctly different time periods based on the type of packet a node is trying to send when the node senses a collision.

The data link layer has more data coming into it as can be seen in Figure 3.2. The collision detect variable is used more than in Ethernet. This does not represent a change in the interface but is a reflection of changes made to the physical layer. The extra information is used in the modified deference procedure. The transmit packet variable in the layer 2-3 interface has one state added to it so that the data link can signal to the third layer when it has started packet transmission.

4.2.4. Physical Layer

No attempt is made to simulate the detail of the electronic behavior of the physical layer. Bits or bytes of a packet are put into a vector of memory that represents the coaxial cable near each node. Nodes receive and pass on the data from left or right neighbors by this vector. A node can receive the following: quiet, zero, one, or collide from the vector that represented the cable. What it passes on to other nodes is a function of its transmit status and the data it receives. What is passed up to the data link layer differs slightly from Ethernet since collision detect is turned on any time a collision is detected, not just during transmission and collision. The physical layer puts the usual preamble onto an outgoing packet and strips it off the incoming packets. All of this represents normal Ethernet behavior at a descriptive level. It differs from the real behavior since it is not passive as an Ethernet tap is and there is no attempt at simulating the attenuation of voltage that goes on as messages passes down the cable. The voltage change caused by the Manchester encoding is not represented either. The additive effect on voltage received from a collision is represented by passing on a "collision" rather than a zero or a one or some combination of garbled bits. Representing the cable signal as bits rather than a continuous function makes this a discrete event simulation.

The discrete time aspect of the simulation is caused by representing transmission on the cable as bits. The bits are represented as the minimum unit of time. Later, in the trial runs, it was found that bytes can be represented as the minimum time unit without altering the results of a long term run. No way has been found to pass packets as a whole since too many events that change the simulation's behavior can happen within a packet's transmission time.

4.2.5. Interfaces

The interfaces, as previously mentioned, are essentially those of Ethernet. The allowed values of the transmit packet variable between the data link layer and the third layer are expanded to allow level three to decide if a packet presented to the data link layer by the third layer has started transmission. This allows layer three to trade a priority packet for an Ethernet packet in

the queue. The type field of a packet is used to specify type 0 for an Ethernet packet and type 1 for a priority packet. As has been previously mentioned, the interface to the physical layer has a jam variable added to it.

4.2.6. Measurement of the Simulation

The statistics that can be collected about the nodes or overall system are of several types. Tobagi provides a taxonomy of methods and types [24]. The following uses his classification technique. While the system was being built and debugged, *snapshots* of system state and node states were generated. These were used to pin point the network's state when a particular event was just about to happen or had happened. *Trace* statistics were used to follow packets and jams as they rippled up and down the net. These measures showed whether a node was transmitting or receiving and what exactly was being sent or received.

Cumulative statistics show long term behavior of the network. Layer 3 collects this type of data for each node. At the end of a simulation run, the data collection module summarizes the data for the whole network. The cumulative data collected on each node is:

1) sum of bytes sent and received for both priority and regular packets,

2) sum of squares of number of bytes sent and received for both priority and regular packets,

3) the number and type of packets sent and received,

4) sum of the delay for received priority and regular packets,

5) sum of squares of the delay for received priority and regular packets,

5) the maximum packet size, packet delay, and collisions encountered by packets of each type.

6) sum of collisions for transmitted priority and regular packets,

7) sum of squares of collisions for transmitted priority and regular packets,

8) the maximum packet size, packet delay and collisions encountered by packets of each type.

The sums of bytes, delays, packets and collisions combined with the amount of time simulated allow the calculation of average throughput/second and average collisions/packet. The sum of squares data is used to calculate the variance in the frequency of collisions and delays. Maximums for delay and collisions are recorded to show the most extreme conditions reached. The delay for this purpose is defined as the difference between the time the packet was queued at its transmitting node and the time of its arrival at the data link layer of the receiver. The delay is made up of delay while waiting in the queue for the second layer to transmit it. The second layer may be deferring or its transmissions may be colliding. Delay is also add as the packet travels from the transmitter to the receiver. Included in this time is the time to pass through the physical layers of the transmitting and receiving nodes.

These measures fulfill two purposes. Some of the measures are used to perform error checking on the system. Total collisions, packet size and total packets sent and received are measures that allow error checking. For example, if total collisions exceeds sixteen for an Ethernet packet, or one for a priority packet, the system is in error. Another example of error checking is packets sent and received. If the totals do not match, some nodes are not receiving jams properly.

The second purpose of the cumulative statistics is to measure the system's performance. The crucial statistics for measuring a network's performance are the various aspects of delay and throughput. As Tropper says:

> The modeling of local networks has concentrated on the performance of the communications subnetwork. The traditional measure of efficiency employed for the performance of a network has been that of time delay vs throughput. [26]

These measures are stated relative to system configuration, number of nodes, and packet length. Network performance changes are a multidimensional problem. The demand for service by nodes can vary in length of messages and frequency of service.

Collisions are a feature of Ethernet and of the proposed new protocol. The data link layer software of a node counts the number of collisions in order to know how much to back off. When a packet is successfully transmitted in this simulation, the number of collisions that the packet

suffers is one of the measures collected on the packet. Nodes record the total number of collisions suffered by each type of packet and the highest number of collisions that any one packet experiences.

The collisions are counted for two reasons. Collisions are indicative of the ease with which a packet is transmitted. A greater number of collisions is an indication of higher delay. The number of collisions is also indicative of how close to nondelivery a packet came. Ethernet quits trying to send a packet if the packet has more than sixteen collisions. Since nondelivery is unacceptable in a priority system, the collisions are recorded to show the likelihood of nondelivery for each type of packet. The average number of collisions shows whether the situation occurs often. Recording the number of collisions for regular packets that are competing with priority packets for the coaxial cable also shows if the regular packets are being pushed any closer to nondelivery by the presence of priority packets.

4.2.7. Language Selection

The simulation is done in the C programming language. Simulation languages were not selected to do the simulation because their features are not needed. The ability to represent continuous processes, a feature of many simulation languages, was not needed because no attempt was made to represent continuous processes. The ability to represent parallelism, an other freature of simulation languages, could be adequately done by C. C is a system implementation language. The software constructed is a simulation of system code and C is an appropriate language to do this.

C has some attractive features for simulation of network behavior. For example, a packet is a structure of several numbers, some character data, and a final number. When the packet is transmitted or received it is a stream of bits. The C language has features that allow the overriding of types. The transition of data from numeric and character to a bit stream and then back to numeric and character data can be readily done by over riding types.

The real need in simulating a network is for processing power. What is being represented is several parallel processes, each of which is quite simple. To learn what the long term behavior of a network is, or to learn what the behavior of a large network is, one must simulate many cycles of a process. This can be done by long runs on a uniprocessor or by using parallel processors or by decreasing the resolution of the simulation by simplifying the representation of the object being modeled. Use of a high level simulation language does not significantly help solve any of these problems.

4.3. Experimentation

The simulated network configuration can be rearranged as desired since the setup software read in the load for and location of each node. Four general situations are tested with the simulator. First the simulation is validated by comparison with other available data to see if the simulator behaves like an Ethernet. It also shows whether or not the literature reports are accurate about Ethernet. Simulations were run to compare to Shoch's real network, Blair's simulation and Stuck's analysis [1,17,20].

Second, a simulation that runs a byte-at-a-time is compared to the one that runs a bit-at-a-time to see if the difference is significant. The byte-at-a-time simulation offers an eight fold speed up in the rate of simulation. Since the simulator takes about two hours of VAX11-780 cpu time to mimic a second of simulated time for the bit-at-a-time experiments, speed up was relevant.

Third, the simulator is run at various loads to test the proposed new protocol against Ethernet in several ways. Each load is run as:

1) a pure Ethernet;

2) a mix of priority and Ethernet nodes with equal length packets and;

3) as a mix of priority and Ethernet nodes with longer priority packets.

The arrangement of nodes is, at first, set with equal and maximally wide spacing of priority nodes. The Ethernet nodes are interspersed between the priority nodes. In these trials, the load

put on the network varies between five and sixty percent of network capacity. Higher loads are not used because by the sixty percent load the trends of the system are quite apparent and it was felt that real systems would seldom be called upon to go beyond this level. Randomly generated arrangements of nodes on the cable and various starting seeds for the packet generator were also used. Finally, the priority nodes are randomly arranged on the net with the net kept at maximum length and nodes equally spaced. This is done to check that use of the regular pattern of arrangement of the nodes is not causing artifacts.

Chapter 5

Simulation Results

This chapter presents the results of the simulation runs in four parts. The first part discusses the validation of the simulation. The second part compares the performance of pure Ethernet and the priority system. Changes that the priority system causes in speed of delivery and the number of collisions encountered are outlined. The third part looks at the performance of nonpriority packets compared to priority packets running within the same environment where the nonpriority packets may be forced to defer to the priority packets. The last part of the chapter looks at comparisons between whole systems. An Ethernet and a priority system with equivalent packets are compared and a priority system where the priority packets are longer is also compared to the first two.

5.1. Experimental Validation

The network software simulation was run at a variety of loads and configurations to see if it matched the behavior reported for Ethernet in the literature. The literature offers a variety of studies with which the simulation can be compared [1,17,20]. The simulation ran within the bounds suggested by all these studies. It was never observed to behave in a way that was outside the limits set by the Ethernet protocol [28]. Exact comparison is a bit vague since the data in the papers are graphic and often do not give enough resolution to make an exact comparison. Another problem with exact comparison is that none of the studies can be duplicated. Let us look at them one at a time.

Shoch's study is important because his data was observed on a real system, and averaged over a long time period [17]. His work attempts to measure reality. Shoch observed that heavily loading the network with long packets to 95% of network capacity was possible and that the network was not unstable. In the simulation, if the load assignment for the nodes exceeded the

network capacity, the network did not become unstable, transmission went up to a high and stable rate of 96.4% of capacity for 1500 byte packets with 16 nodes on a 186 m network. This is not as many nodes as Shoch ran but otherwise matches his network. Shoch's reports of collision rates correspond to the results recorded for the simulation. Shoch's network had an average load of 0.80% and had 0.0003 collisions/packet. The simulation has 0.01 collisions/packet at the five percent level. The rate of collisions for the simulation decreases by a factor of three every time the load is halved. At 0.8% the collision rates are within 0.00015. Duplication of the loading by the real system is not possible since the simulation would have to run several years. Another problem is reproducing the distribution of packet sizes peculiar to the Xerox net, including such items as spikes for popular games. Shoch reports nothing on delay and nothing on network behavior with short packets and heavy loading, which are the weak points of Ethernet.

Blair's simulation study is easily compared to my simulation. His study compares the Cambridge Ring and Ethernet. In doing the study, Blair forced similarities between Ethernet and Cambridge Ring. A problem arises in comparing Ethernet to other systems since it is so thoroughly specified. Blair's reported delays for the same loads are the same as the delays observed by running my simulation with that load. Blair for a 16 node network with a one kilometer length and five, ten and twenty percent loads reports the same average delay/packet as my simulation had running over twice the distance (see Figure 5.1). This appears to be caused by Blair not exactly following the Ethernet protocol and imposing an acknowledgement feature that is not part of Ethernet. He also deviated from the specified packet size in some of the runs making the packets as short as 16 bytes. The trend of Blair's results is, however, exactly duplicated by my simulation. Both simulations indicate very low delay of less than 300 μs per packet at five percent load, with the delay accelerating rapidly after the load reaches 20% of capacity. Blair's delay versus load curves change in the same way at the same points as my simulation.

Stuck's paper is a mathematical analysis of the performance of various broadcast and ring nets at high load and varying capacities. He offers a formula for calculating the delay of Ethernet

but omits the detail of the collision factor, making calculated checks of the simulation impossible. His paper predicts that Ethernet can have throughputs that are in excess of 90% of capacity, the observation made by Shoch for high load and also observed with my simulation.

The results of my simulation of Ethernet corresponds to the results of the above studies. Its behavior, by tracing the activity of the net as packets were sent and received, is as the Ethernet specification sets out. The simulation is considered valid.

Once the basic software was checked a second version that gave a time resolution of a byte, not a bit, was tried. Hence time resolution was coarser by a factor of eight. The variation in average delay after a trial run of a few thousand packets (1 s) was less than one percent. Other statistics were similarly undisturbed. From then on the simulation with byte resolution was used.

One further preliminary item to be defined is how long to simulate the network to erase the random effects of short samples. Differences in average load and delay drop to less than one percent when one second of time is simulated. The test runs were done for ten seconds of simulated time.

5.2. Priority vs Ethernet

The design of the proposed protocol leaves little doubt that the priority packets will get a better average and maximum response time with fewer collisions than the Ethernet packets. The simulation results give quantitative values to the improvements in performance. The maximum worst cases experienced are of particular interest because they can not be analytically determined. The theoretical worsts are horrible but not probable. The simulation provides numeric values that give a better feel than available analytic methods for what the performance would be like in reality.

The first four figures of this chapter show two major benefits of the priority system as a means of getting messages quickly and predictably to their destination. One effect is on average packet delay, the other is on maximum packet delay. The figures also show that the effects of

loading falls into two zones, the low load zone below 40% load and the high load zone above 40%. Both the effects on packet performance show up when the net's throughput is in the high load zone. Only the effect on maximum delay occurs at lower load levels.

Figure 5.1 graphs the average delay time of a packet in the two systems. Packets are 128 bytes long for Ethernet and priority. The network is 16 nodes on a 2.5 km cable running at 10 Mbits/s. One quarter of the 16 nodes in the priority environment were priority nodes. The X-axis represents the throughput as a percent of the total available capacity. It can be translated to Mbit/s by dividing by ten. To generate each point of the graph, ten seconds of network time were simulated. Table 5.1 gives the numbers from which this graph was constructed.

The first and most important effect pointed out by the simulation is on the average delay that a packet encounters. For a packet transmitted in an environment of 60% load, the average delay of a priority packet is thirty times less, (about 650 ms compared to 1.7 ms), than for an Ethernet packet. Where the Ethernet has a typical higher and accelerating delay starting at the forty percent load level the priority scheme does not, it continues with a steady slope, (Figure 5.1). This effect rapidly disappears as system load decreases since the delay on the Ethernet is an exponential function of load. At low loads the benefit of the priority scheme is not apparent from the average delay since virtually all packets are transmitted without collisions. This points to a strength of the new protocol since very often the priority packets being sent as Ether packets get sent faster than they could be by any other scheme. At less than 20% load the priority packets do no better, on the average, than Ethernet packets.

The maximum delay graph, Figure 5.2, is the worst that happened to a packet. Figure 5.2 gives the maximum delay time of a packet in the two systems. The environments for the packets were the same as the environments for the last graph. A table of numbers for generating the graph is in Table 5.1. A priority message should be guaranteed a maximum delay that is close to the average delay and has a ceiling. The maximum delay experienced for priority packets even at low loads is 1/8 that of Ethernet. As system load rises, the priority packets advantage increases.

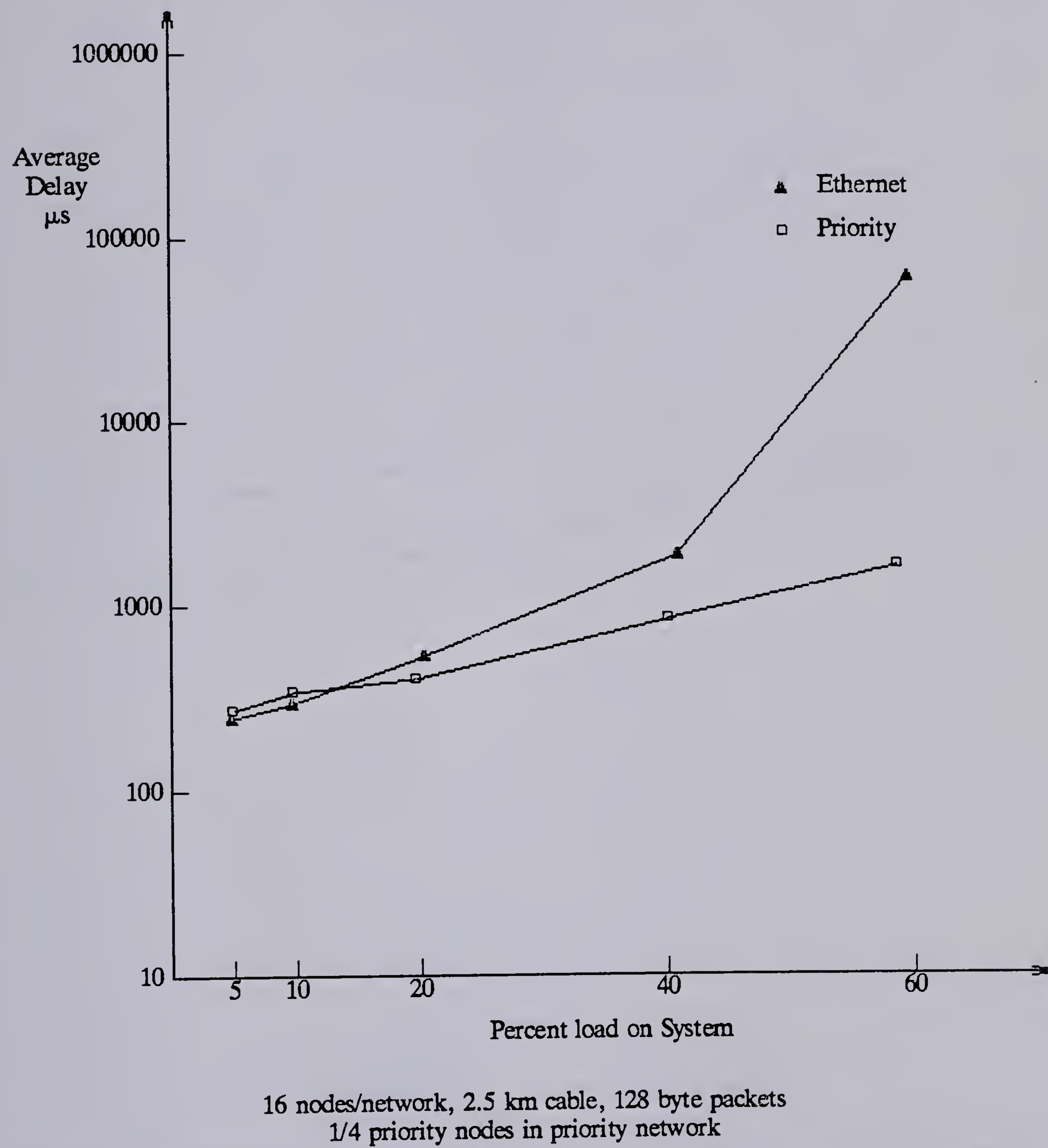
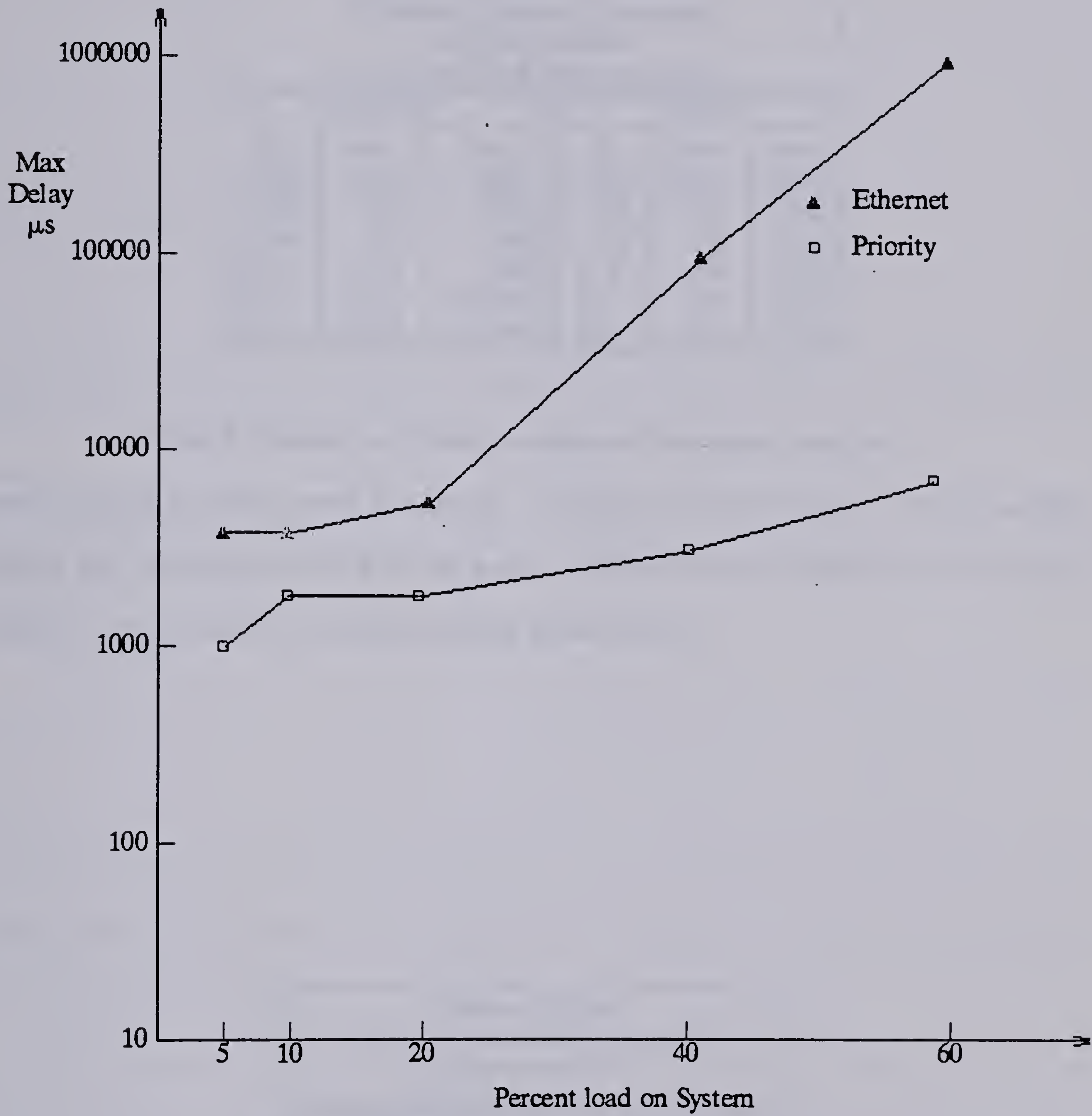


Figure 5.1 Ethernet vs Priority: Average Delay Time

By sixty percent load it is 25 times better than Ethernet.

Table 5.2 gives a table of the variance of the delay times for the packets in the Ethernet and priority systems. The variance of delay for priority packets is about the same as Ethernet at a five



16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in priority network

Figure 5.2 Ethernet vs Priority: Maximum Delay Time

Data for Figures 5.1 and 5.2 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority					
Ethernet			Priority		
% load	ave delay	max delay	% load	ave delay	max delay
4.92	220	3810	4.93	233	992
9.74	302	3840	9.74	343	1860
20.4	552	5490	19.6	405	1830
40.2	1960	101000	40.0	875	3220
59.5	64769	1.8×10^6	58.5	1712	7270

Table 5.1 Ethernet vs Priority: Average and Maximum Delay in μs

percent load but is quickly passed by Ethernet. The priority packet's variance is 10^4 less at 60%, meaning that the priority system gives far greater predictability in its behavior. It is thus quite suitable for use as a means of reliably expediting packet delivery.

Variance of Delay 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority			
Ethernet		Priority	
% load	var delay	% load	var delay
4.92	9.91×10^4	4.93	9.78×10^4
9.74	3.10×10^5	9.74	3.30×10^5
20.4	1.27×10^6	19.6	2.98×10^5
40.2	2.44×10^7	40.0	1.06×10^6
59.5	4.95×10^{10}	58.5	1.45×10^6

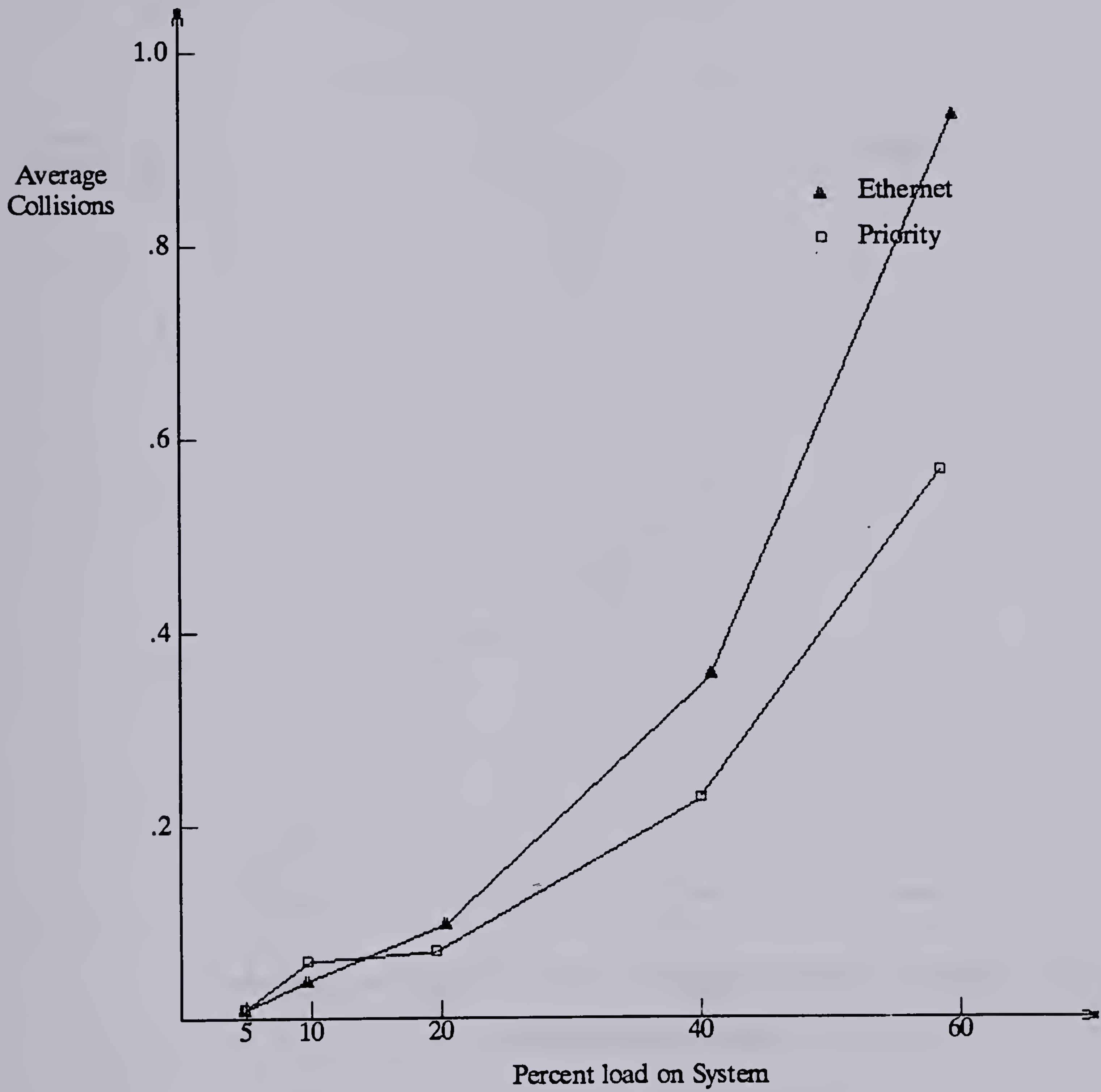
Table 5.2 Ethernet vs Priority: Variance of Delay

Figures 5.3 and 5.4 show the average and maximum collision rates for packets. A table of the data points is in Table 5.3. Packets are 128 bytes long for regular and priority. The network is 16 nodes on a 2.5 km cable running at 10 Mbits/s. One quarter of the 16 nodes in the priority environment were priority nodes giving off priority packets. The X-axis represents the throughput as a percent of the total available capacity. It can be translated to Mbit/s by dividing by ten. To generate each point of the graph, ten seconds of network time were simulated.

The average number of collisions experienced is of interest since it affects delay. Figure 5.3 does not indicate this, but if the loading was studied, beyond sixty percent the Ethernet packets would suffer ever increasing numbers of collisions until they reached an average of $\log_2$ nodes on the net. In this case $\log_2 16 = 4$ collisions. The priority packets would approach an average of one collision as the load rose regardless of network size or number of priority nodes. This one collision is far below $\log_2$ nodes, four, in this case.

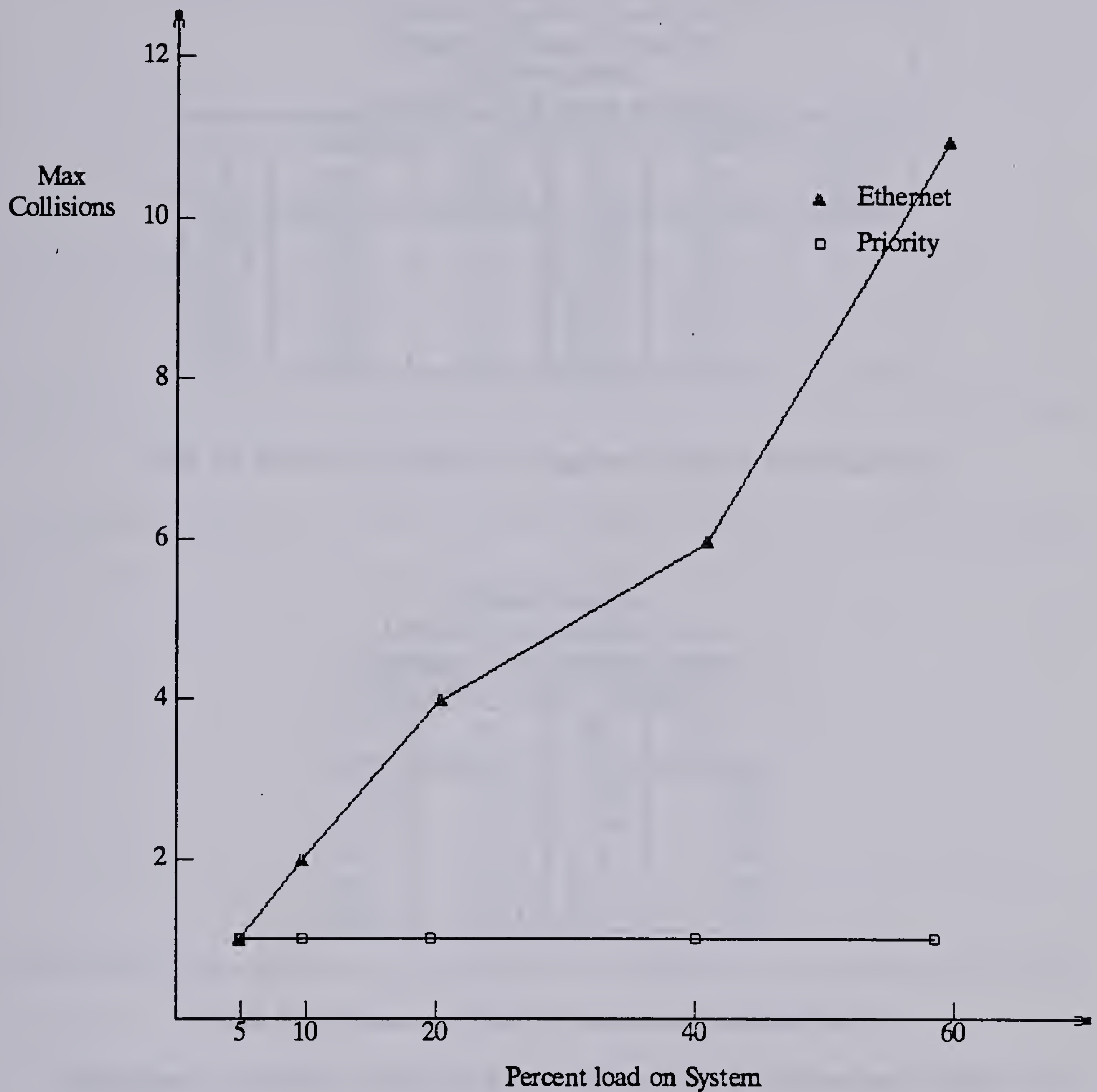
Maximum collisions, shown in figure 5.4, are of interest because on Ethernet, if the number of collisions gets as high as 16 for a particular packet, the transmitting node quits trying to send the packet. The simulation done is of a small population, 16 nodes, and yet some packets have 11 collisions at only a 60% load level. Nondelivery is not acceptable for a priority system. Nondelivery would certainly occur at higher loads and seems likely, even at this load, if the system ran long enough. The priority scheme has effectively dealt with this problem.

Table 5.4 shows the variance of collisions for the packets in both the Ethernet and the Priority environments. Again the priority system has the lower variance, indicating a more predictable performance at high loads. At low loads the Priority system has more variance because the node intentionally rebroadcasts immediately after a collision. Wider variance of the collision rate at such a low average collision rate in not a serious defect.



16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in priority network

Figure 5.3 Ethernet vs Priority: Average Number of Collisions/Packet



16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in network

Figure 5.4 Ethernet vs Priority: Maximum Number of Collisions/Packet

Data for Figures 5.3 and 5.4 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority					
Ethernet			Priority		
% load	ave collisions	max collisions	% load	ave collisions	max collisions
4.92	0.01	1	4.93	0.01	1
9.74	0.03	1	9.74	0.06	1
20.4	0.11	4	19.6	0.07	1
40.2	0.43	6	40.0	0.23	1
59.5	0.93	10	58.5	0.55	1

Table 5.3 Ethernet vs Priority: Average and Maximum Collisions/Packet

Collision Variance 10 Mbits/s 16 nodes 2.5 km cable for priority 1/4 of nodes on priority			
Ethernet		Priority	
% load	var collisions	% load	var collisions
4.92	0.01	4.93	0.02
9.74	0.05	9.74	0.07
20.4	0.11	19.6	0.07
40.2	0.57	40.0	0.22
59.5	1.81	58.5	0.27

Table 5.4 Ethernet vs Priority: Variance of Collisions/Packet

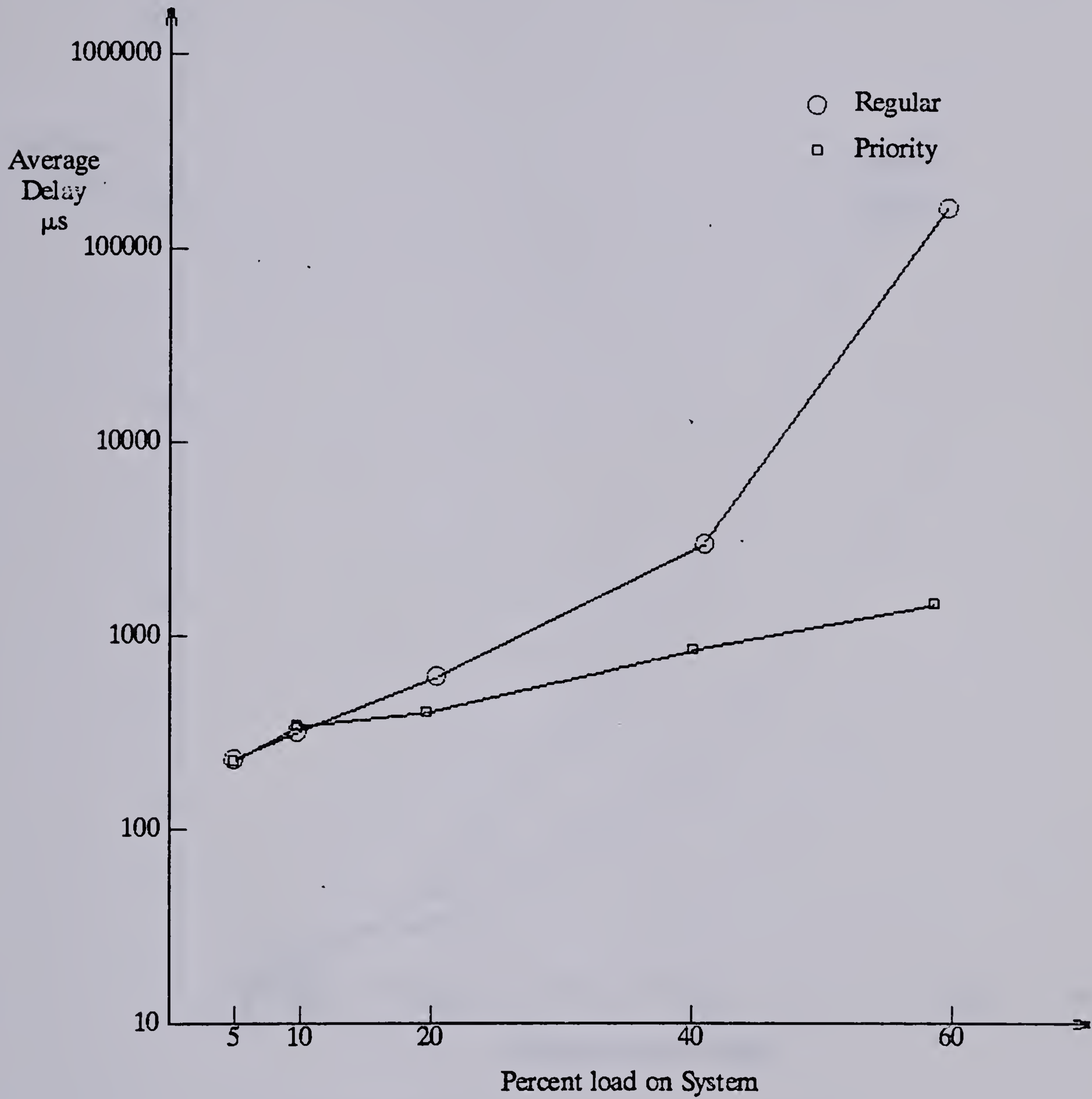
In summary, the priority system works well. At low loads it keeps maximum delay to 1/4 of the Ethernet. At high loads it gives delivery with only 1/30 the average delay of Ethernet. The previous observations clearly show that the new protocol gives improvements in the delay characteristics of the packets delivered with a priority status in the new protocol compared with packets delivered by a regular Ethernet. These gains in the new protocol affect only part of the traffic on the network. The rest of the traffic has the usual delay of Ethernet packets plus extra delay as they defer to priority packets. The following results show the effect of this arrangement.

5.3. Priority vs Nonpriority Packets, Same System

Figures 5.5, 5.6 and 5.7 compare the average delay, maximum delay and collisions of priority packets to nonpriority packets competing with each other in the same environment. Packets are 128 bytes long for both regular and priority service. The network is 16 nodes on a 2.5 km cable running at 10 Mbits/s. One quarter of the 16 nodes are priority nodes. All the benefits to priority packets mentioned in the Ethernet vs priority comparison show up in the comparison between priority and nonpriority packets in the same system. The effects are magnified since the regular packets have to defer to the priority packets. The tables of figures from which the graphs were generated are presented in Tables 5.5 and 5.5.

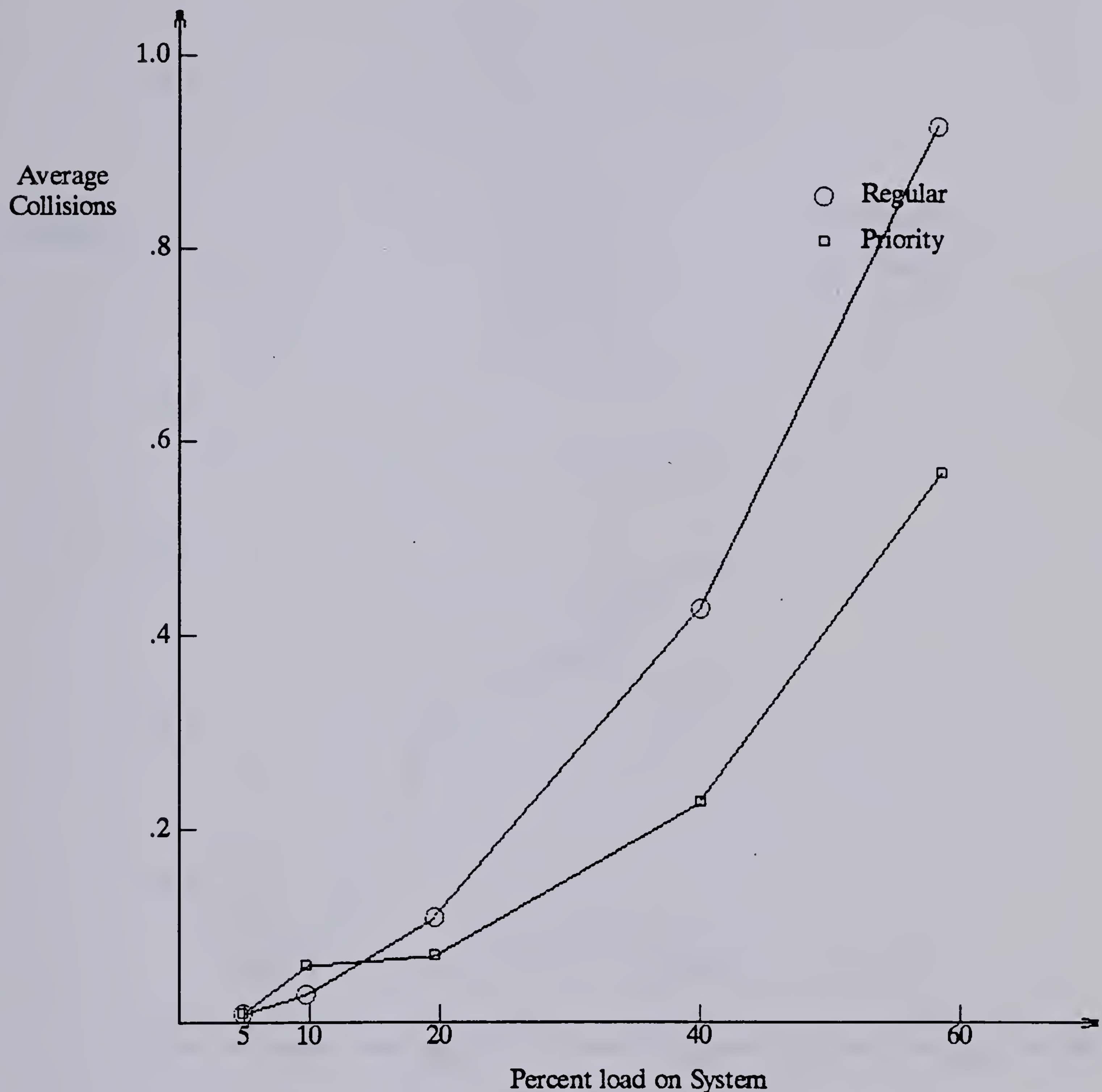
Figure 5.7 shows that the priority nodes were behaving properly and not experiencing more than one collision. It also shows a far more important aspect of the behavior of the network, the priority packets were not causing an increase in numbers of collisions for the regular packets. Referring back to Figure 5.4, we can see that the curve for maximum collisions for regular packets is virtually identical to the curve of collisions for Ethernet packets. Nonpriority packets are not in any greater jeopardy of nondelivery when the priority packets are added to the system load.

Tables 5.7 and 5.9 show the variance data on the priority and regular packets for delay and collisions respectively. The priority packets have the best performance.



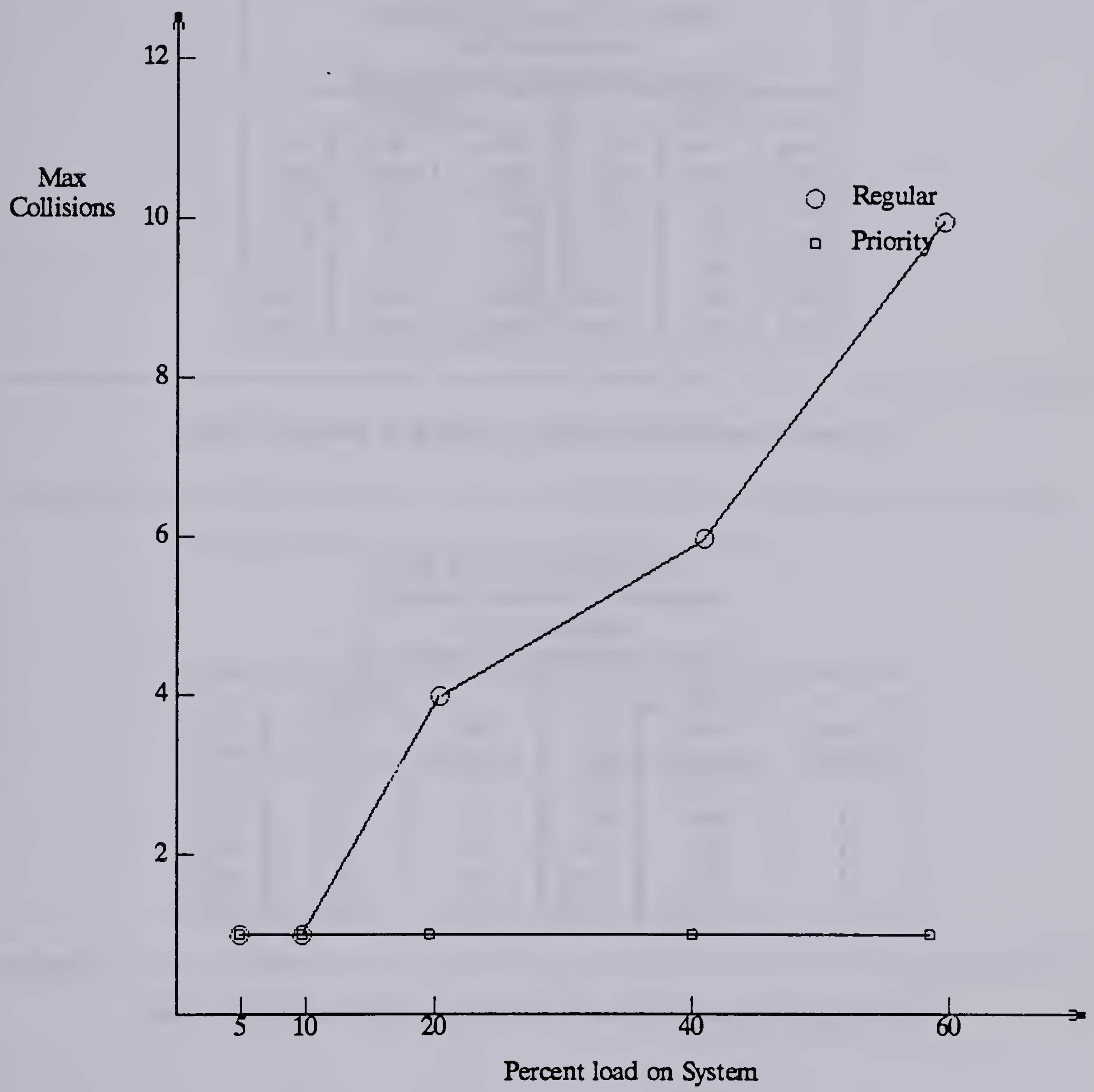
16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in network

Figure 5.5 Regular vs Priority: Average Delay Time



16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in network

Figure 5.6 Regular vs Priority: Average Collisions/Packet



16 nodes/network, 2.5 km cable, 128 byte packets
1/4 priority nodes in network

Figure 5.7 Regular vs Priority: Maximum Collisions/Packet

Data for Figure 5.5 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority					
Regular			Priority		
% load	ave delay	max delay	% load	ave delay	max delay
4.93	233	3810	4.93	233	992
9.74	330	2400	9.74	343	1860
19.6	633	8290	19.6	405	1830
40.0	3098	48800	40.0	875	3220
58.5	38200	2.1×10^6	58.5	1712	7270

Table 5.5 Regular vs Priority: Average and Maximum Delay in μs

Data for Figures 5.6 and 5.7 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority					
Regular			Priority		
% load	ave collisions	max collisions	% load	ave collisions	max collisions
4.93	0.01	2	4.93	0.01	1
9.74	0.03	1	9.74	0.06	1
19.6	0.11	4	19.6	0.07	1
40.0	0.43	6	40.0	0.23	1
58.5	0.96	9	58.5	0.57	1

Table 5.6 Regular vs Priority: Average and Maximum Collisions/Packet

Variance of Delay 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority			
Regular		Priority	
% load	var delay	% load	var delay
4.93	1.57×10^5	4.93	9.78×10^4
9.74	3.89×10^5	9.74	3.30×10^5
19.6	2.28×10^6	19.6	2.98×10^5
40.0	5.21×10^7	40.0	1.06×10^6
58.5	7.21×10^{10}	58.5	1.45×10^6

Table 5.7 Regular vs Priority: Variance of Delay

Variance of Collisions 10 Mbits/s 16 nodes 2.5 km cable 128 bytes packets for priority 1/4 of nodes on priority			
Regular		Priority	
% load	var collisions	% load	var collisions
4.93	0.01	4.93	0.02
9.74	0.03	9.74	0.07
19.6	0.17	19.6	0.07
40.0	0.72	40.0	0.22
58.5	1.90	58.5	0.27

Table 5.8 Regular vs Priority: Variance of Collisions/Packet

5.4. Whole System Comparisons

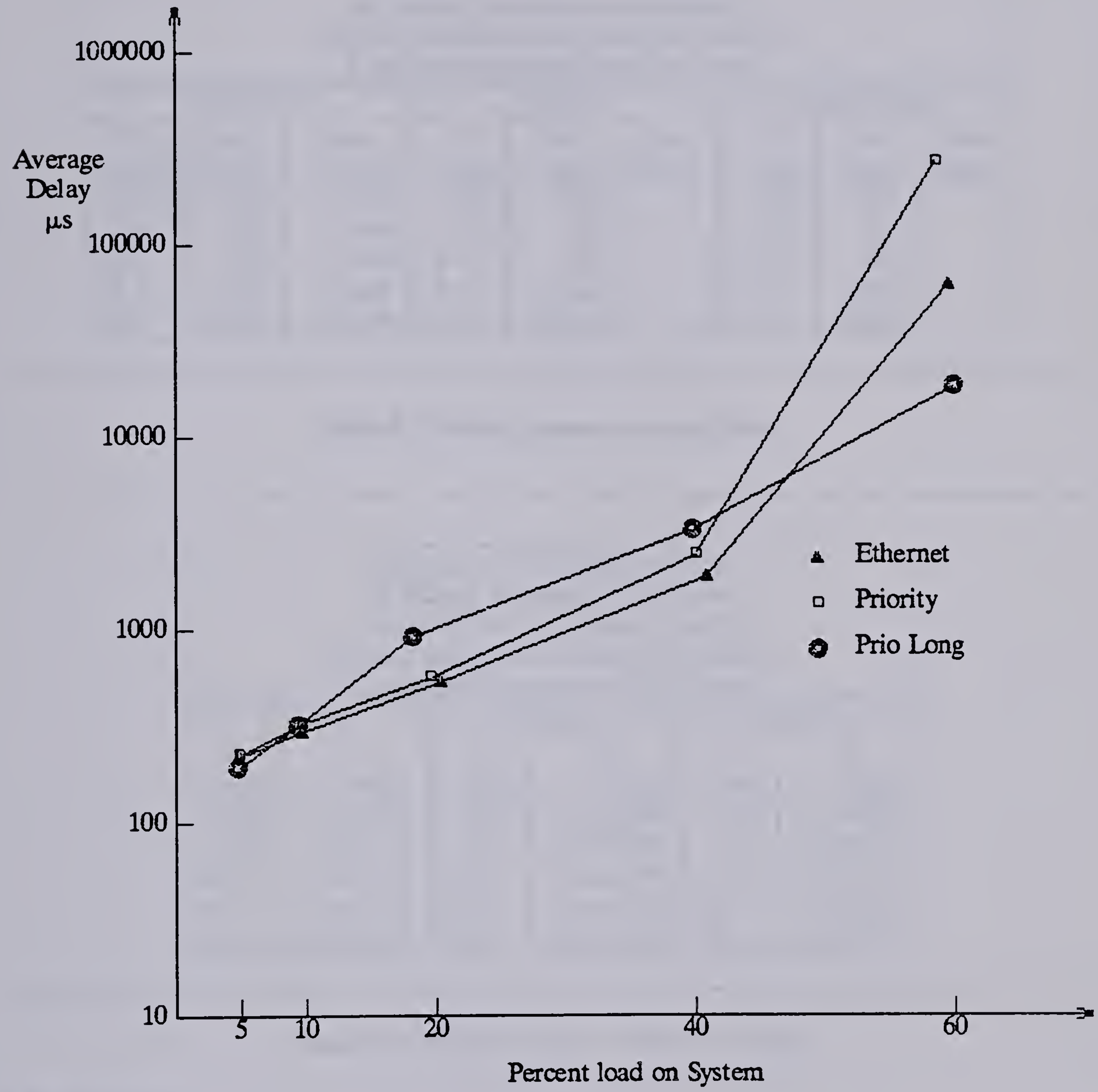
Figure 5.8 shows the average delay for all packets in an environment. Packets are 128 bytes long for Ethernet and Priority Short environments, the ones discussed to this point. Priority packets are 1500 bytes long for priority packets only in the Priority Long environment, the other 75% of the packets are regular packets 128 bytes long. The network is 16 nodes on a 2.5 km cable running at 10 Mbits/s. One quarter of the 16 nodes could send priority packets. The rest of the traffic in each of the priority environments was 128 byte regular packets. Table 5.9 is the numeric data

for generating Figure 5.8.

The use of priority packets in a system has a cost. The other packets defer to the priority packets increasing their own delay by doing so. Only if the increased delay to the regular packets is acceptably small, would such a system be used. The question of how great the cost to the whole system is for priority is relevant at high loads since the delay is greatest then. The difference in placement of the Ethernet graph line and the Priority line in Figure 5.8 show how the priority packets affect the whole system. It can be seen that, up to 40% load, the difference in delay is less than 1.0 ms. However, at 60% to provide priority, the cost is 200 ms per packet or a factor of four. The whole of the cost in delay to the system is paid for by the nonpriority packets. Whether this delay increase of 0.2 s is acceptable would depend on the system application.

The third graph line, Priority Long, on Figure 5.8 shows the effect of having the priority nodes give out larger messages, or as if 10 short messages were bunched together. The effect of this is beneficial at high load where average delay for a long priority packets and regular packets is cut to one-quarter the delay of an Ethernet packet. If the use being made of the nodes and network permitted this, it would be very attractive. A computer responding with a burst of several messages to several terminals would cause this workload.

Table 5.10 gives the variance on the average delay for the three systems graphed in Figure 5.8. Overall the variance for a whole system with 25% of the load being priority packets is worse than for the pure Ethernet. The system with the long priority packets has very little variance by comparison to an Ethernet. If a system could be configured like the long priority system it would undoubtedly be beneficial to implement the priority feature.



16 nodes/network, 2.5 km cable
128 byte Ethernet and Priority packets
1500 byte Long Priority packets
1/4 priority nodes in priority networks

Figure 5.8 Whole Systems: Average Delay all Packets

Data for Figure 5.8 10 Mbits/s 16 nodes 2.5 km cable for priority 1/4 of nodes on priority 128 bytes packets for Ethernet and Priority Long Priority packets are 1500 bytes								
Ethernet			Priority			Long Priority		
% load	ave delay	max delay	% load	ave delay	max delay	% load	ave delay	max delay
4.92	220	3810	4.93	232	-	4.76	197	-
9.74	302	3840	9.74	333	-	9.33	324	-
20.4	552	5490	20.4	557	-	18.2	954	-
40.2	1960	101000	40.0	2547	-	39.7	3364	-
59.5	64769	1.8×10^6	58.5	281531	-	58.6	19000	

Table 5.9 Whole Systems: Average Delay

Variance of Whole System Delay 10 Mbits/s 16 nodes 2.5 km cable for priority 1/4 of nodes on priority 128 bytes packets for Ethernet and Priority Long Priority packets are 1500 bytes					
Ethernet		Priority		LPriority	
% load	var delay	% load	var delay	% load	var delay
4.92	9.91×10^4	4.93	1.40×10^5	4.76	1.10×10^5
9.74	3.10×10^5	9.74	3.75×10^5	9.33	1.04×10^6
20.4	1.27×10^6	19.6	1.80×10^6	18.2	9.03×10^6
40.2	2.44×10^7	40.0	4.04×10^7	39.7	8.59×10^7
59.5	4.95×10^{10}	58.5	4.52×10^{11}	58.6	5.26×10^9

Table 5.10 Whole Systems: Variance of Delay

5.5. Summary

Clearly, the performance of the priority packets in the network is different from the performance of the Ethernet packets in the network. The priority packets suffer fewer collisions and endure far less delay. The worst case results for the priority packets are far better than the Ethernet packets. The worst case results for the priority packets are also far closer to the average case behaviors for their type of packets. All of these improvements mean that the implementation of a

priority feature for the network is far more feasible than is the case for regular Ethernet.

Chapter 6

Conclusions and Further work

6.1. Conclusions

6.1.1. Overview

This study makes the following contributions to knowledge on networks. A method of increasing the information available to a node on a broadcast network is explained. A protocol that implements the use of priority packets on Ethernet is also developed. The value of such a network approach is exposed by a simulation of the network.

6.1.2. Information Transfer

This study has shown that receivers can sense more than a stream of bits on the broadcast medium of the Ethernet. The other information that can be sensed is a voltage level which indicates the occurence of collisions and time periods from when signals or collisions or silence start until some new state occurs. The voltage and time for collisions is information on the network's state and can be used to trigger behaviors of individual nodes in building network protocols. In the simulation study, the extra information is used to add a priority feature to Ethernet. The simulation runs, discussed in Chapter Four, switches a network from an Ethernet to a token bus successfully. The removal of some of the random behavior of the Ethernet, guarantees delivery of packets, something which Ethernet does not do. Both the random behavior that determines who will signal next, and the random choice of a restart time by a node is altered in the simulated network. For such a network, there is a penalty in overall system delay and ultimately throughput because of the addition of the priority feature.

In the system developed, priority can only be provided for nodes that have been configured with this ability. The performance gain in reduced delay is negligible at low network loads but

quite good as system load rises above forty percent of theoretical capacity. The provision of this priority feature for an Ethernet is not complex. The very minor changes to the transceivers were discussed in Chapter Three. Internode connection is still baseband coaxial cable, and elaborate collision avoidance is not used. Nodes are more influenced by information sent by other nodes but the network still has distributed control.

Analysis of the performance of a protocol using this information shows that priority packets can have their delay bounded by an upper limit which is linearly proportional to the number of nodes able to generate such packets. An illustration of the benefit of the simulated protocol is that priority packets suffer less than one percent as much delay as regular Ethernet packets at the 60% load level. The implementation of such a feature costs the rest of the packets on the same network an increased average delay of about four times the delay of a regular Ethernet.

Using this new protocol, selected devices can gain a better response time for packet delivery and the opportunity of issuing priority messages. There is a resulting cost in delay to other packets on the system if all packets are the same average size. However, if several small priority packets can be run together by one node and sent as larger units, overall system performance will be improved as is shown in Figure 5.8. An example of where this would find practical use is computers servicing a terminal population. With regular Ethernet, a computer could potentially be bottlenecked as it tried to access the network. The new protocol would allow the overall system a better response time by allowing one node to send a great deal of traffic as a large packet. The priority scheme allows a single device or set of devices greater access to the network. In the computer/terminals example, gaining the extra access to the network would be very beneficial.

6.2. Further Work

Considerable further work can be done on some of the points discussed in the thesis. Trying to further enhance the signaling between nodes, yet retaining the simplicity of a medium such as coaxial cable, would allow work on some elegant use of telecommunications.

If the length and amplitude of a jam carries information, measuring them more precisely may be of benefit to standard Ethernet. Exact measurement of all detected collisions would allow estimation of how many stations had collided and thus how many stations wished to send messages. The back-off algorithm could then use this as the bound for the selection of random restart times. One collision, or at least fewer collisions, would then suffice to solve the contention where several were needed previously. The scheduling of a network would then be speeded up. Less capacity would be wasted on collisions, and overall network performance enhanced. Altered voltages for transmission is such a way of signaling.

A diagnostic node could be developed for a network. Such a node would cause a protocol change to suit general load conditions of the network. This could be done on Ethernet by a promiscuous node which can listen to all traffic. If such a node could detect network use by many users, causing high demand with short average packets, it could transmit a jam or packet to trigger conversion to a new protocol more suited to the demand at that time. This would allow use of Ethernet under conditions where it is strong, and some other system under other conditions. This is essentially an extension of allowing a node to diagnose its need for priority. The promiscuous node diagnoses the need of the system and calls for a protocol to suit this need.

The protocols that switch between major behavior patterns are a subject needing further work. Particular methods would suit particular situations but a definition of such situations has not been done.

Broadcast baseband networks can be developed beyond their current state. This will however require more sophisticated transceivers to pass messages other than bits via the transmission media. The protocols used on such systems, if they are combinations of the presently known techniques, will allow the network to adapt to the momentary needs of the network and its nodes. Such protocols would allow even better allocation of network resources than is now possible. Ultimately construction of prototype of the network needs to be undertaken to test predictions made by mathematic analysis and simulation.

References

[1] G S Blair and D Shepherd, A Performance Comparison of Ethernet and the Cambridge Digital Communication Ring, *Computer Networks 6*, 2 (May 1982), 105-114.

[2] R P Blanc, Off the Shelf Solutions Motivate NBS's Standards Drive, *Computer Design 21*, 3 (March 1982), 21-23.

[3] D Bursky, PLA's, Software Compiler Give Ethernet Controller Highly Structured Design, *Electronic Design 30*, 18 (September 2, 1982), 35-36.

[4] J I Capetanakis, Generalized TDMA: The Multi-Accessing Tree Protocol, *IEEE Trans Comm COM-27*, 10 (October 1979), 505-514.

[5] L Carroll, *Alice in Wonderland and Other Favorite Stories*, Pocket Books, New York , 1951.

[6] I Chlamtac and W R Franta, BRAM: The Broadcast Recognizing Access Method, *IEEE Trans Comm COM-27*, (August 1979), 1183-1190.

[7] I Chlamtac and W R Franta, Performance Issues in Backend Storage Nets, *Computer 13*, 2 (February 1980), 18-33.

[8] 3COM Corp, *UNET Network Software Brochure* , 3COM Corp , 1981.

[9] R Crane, Ethernet Designer's Guide, *Microprocessors and Microsystems 6*, 8 (October 1982), 405-412.

[10] D Elliot and S Kopec, One Chip Carries out Ethernet Protocol, *Electronic Design 30*, 20 (September 30, 1982), 121-132.

[11] W R Franta and I Chlamtac, *Local Networks*, Lexington Books, 1981.

[12] Y I Gold and W R Franta, An Efficient Collision-Free Protocol for Prioritized Access-Control of Cable or Radio Channels, *Computer Networks 7*, 2 (April 1983), 83-98.

[13] L W Hansen and M Schwartz, An Assigned-Slot Listen-Before-Transmission Protocol for a Multiaccess Data Channel, *IEEE Trans Comm COM-27*, 6 (June 1979), 846-857.

[14] L Kleinrock, *Queuing Systems: Computer Applications*, Wiley Interscience, 1975.

[15] F Kuo and N Abramson, *Computer Communication Networks*, Prentice-Hall, 1973.

[16] R M Metcalf and D R Boggs, Ethernet: Distributed Packet Switching for Local Computer Networks, *Comm. ACM 19*, 7 (July 1976), 395-404.

[17] J F Shoch and J A Hupp, Measured Performance of an Ethernet Local Network, *Comm. ACM 23*, 12 (December 1980), 711-721.

[18] J F Shoch, Evolution of the Ethernet Local Computer Network, *IEEE Computer 15*, 8 (August 1982), 10-28.

[19] O Spaniol, Modeling of Local Computer Networks, *Computer Networks 3*, 5 (November 1979), 315-326.

[20] B W Stuck, Calculating the Maximum Mean Data Rate in Local Area Networks, *IEEE Computer 16*, 5 (May 1983), 72-76.

[21] A S Tannenbaum, Network Protocols, *ACM Surveys 13*, 4 (December 1981), 453-490.

[22] A S Tannenbaum, *Computer Networks*, Prentice-Hall, 1981.

[23] J E Thornton, Back-End Network Approaches, *Computer 13*, 2 (February 1980), 3-11.

[24] F A Tobagi, Modeling and Measurement Techniques in Packet Communication Networks, *Proceedings IEEE 66*, 11 (November 1978), 1423-1447.

[25] F A Tobagi, Multiaccess Protocols in Packet Communication Systems, *IEEE Trans Comm COM-28*, 4 (April 1980), 468-488.

[26] C Tropper, *Local Computer Network Technologies*, Academic Press, 1981.

[27] J White and D Dalal, Higher Level Protocols Enhance Ethernet, *Electronic Design 30*, 8 (April 15, 1982), SS33.

[28] XEROX, The Ethernet: A Local Area Network Datalink Layer and Physical Layer Specifications, *Xerox Digital Intel*, 1980.

[29] B P Zeigler, *Theory of Modeling and Simulation*, Wiley, 1976.